University of Zurich[UZH]

Department of Informatics

Visual Inertial Odometry and Active Dense Reconstruction for Mobile Robots

Dissertation submitted to the Faculty of Business,
Economics and Informatics
of the University of Zurich

to obtain the degree of
Doktor / Doktorin der Wissenschaften, Dr. sc.
(corresponds to Doctor of Science, PhD)

presented by
Christian Forster
from Switzerland

approved in April 2016

at the request of
Prof. Dr. Davide Scaramuzza, advisor
Prof. Dr. Marc Pollefeys, examiner
Dr. Stefan Leutenegger, examiner

The Faculty of Business, Economics and Informatics of the University of Zurich hereby authorizes the printing of this dissertation, without indicating an opinion of the views expressed in the work.

Zurich, April 6, 2016

Chairwoman of the Doctoral Board: Prof. Dr. Elaine M. Huang

To my parents.

Acknowledgements

First and foremost, I would like to thank my supervisor Prof. Davide Scaramuzza for selecting me as his first PhD student. Davide's vision of robotics inspired me, he provided me with plenty of exciting opportunities, and he also gave me the freedom to pursue my own ideas. It was an extremely rewarding experience to see the lab grow in the last 3.5 years from just the two of us to more than a dozen staff members. Collaboration enjoys a very high priority in this lab and indeed, this thesis would not have been possible without the help, fruitful discussions, and fun distractions from my collegues. I therefore wish to express my gratitude to all the current and past members, visitors, and students of the Robotics and Perception Group. I would particularly like to thank Matia Pizzoli, Manuel Werlberger, Guillermo Gallego, Jeffrey A. Delmerico, Reza Sabsevari, Elias Müggler, Matthias Fäessler, Flavio Fontana, Zichao Zhang, Michael Gassner, Henri Rebecq, Junjie Zhang, Davide Falanga, Gabriele Costante, Andras Majdik, Volker Grabe, and Andrea Censi.
Furthermore, I would like to thank Laurent Kneip and Simon Lynen that were excellent supervisors during my Masters thesis at ETH where I first learned about visual SLAM. I am extremely greatful to Luca Carlone and Prof. Frank Dellaert for inviting me to a very inspring stay at Georgia Tech.
I would also like to thank everyone associated to the Artificial Intelligence Laboratory that accompanied me during my early days of my PhD.
I would like to thank the PhD commitee members Prof. Marc Pollefeys and Dr. Stefan Leutengger for accepting to review my thesis and for their very valuable feedback.
Finally, I am most grateful to my girlfriend Alina Kyburz, my family, and my friends who supported me at all times.

Zurich, December 2015 C. F.

Abstract

Using cameras for localization and mapping with mobile robots is appealing as these sensors are small, inexpensive, and ubiquitous. However, since every camera image provides hundred thousands of measurements, it poses a great challenge to infer structure and motion from this wealth of data in real-time on computationally constrained robotic systems. Furthermore, robustness becomes an important factor when applying computer vision algorithms to mobile robots that are moving in uncontrolled environments. In this case, nuisances such as occlusions, illumination changes, or low textured surfaces increase the difficulty to track visual cues, which is fundamental to enable camera-based localization and mapping.

The first contribution of this thesis is an efficient, robust, and accurate visual odometry algorithm that computes the motion of a single camera solely from its stream of images. Therefore, the use of direct methods that operate directly on pixel level intensities is investigated. The advantage of direct methods is that pixel correspondence between images is given directly by the geometry of the problem and can be refined by using the local intensity gradients. However, joint refinement of structure and motion by pixel-wise minimization of intensity differences becomes intractable as the map grows. Therefore, a novel semi-direct approach is proposed that establishes feature correspondence using direct methods and subsequently relies on proven feature-based methods for refinement. We further show how inertial measurements can seamlessly be integrated in the optimization of structure and motion. Therefore, the second contribution of this thesis is a preintegration theory that allows summarizing many inertial measurements between two frames into single relative motion constraints. We formally discuss the generative measurement model as well as the nature of the rotation noise and derive the expression for the maximum a posteriori state estimator. Experimental results confirm that our modeling efforts lead to accurate state estimation in real-time, outperforming state-of-the-art approaches.

Tracking salient features in the image results in sparse point clouds; however, for robotic tasks such as path planning, manipulation, or obstacle avoidance, a denser surface representation is needed. Previous work on dense reconstruction from images aim at providing high fidelity reconstructions. However, for robotic applications, the accuracy of the reconstruction should be governed by the interaction task. Furthermore,

it is crucial to have a measure of uncertainty in the reconstruction, which aids motion planning and fusion with complementary sensors. This motivates the third contribution of this thesis, which is an efficient algorithm for probabilistic dense depth estimation from a single camera. Therefore, we combine a multi-view and per-pixel-based recursive Bayesian depth estimation scheme with a fast smoothing method that takes into account the estimated depth uncertainty.

While most computer vision approaches fuse depth-maps in a cost volume, care has to be taken in terms of scalability and memory consumption for robotic applications. Therefore, building upon the proposed dense depth estimation, the next contribution of this thesis is a robot-centric elevation mapping system that suits a flying robot with down-looking camera and can be applied on-board *Micro Aerial Vehicles* (MAVs) for fully autonomous landing-spot detection and landing.

We further demonstrate the usefulness of dense depth-maps for localization of an MAV with respect to a ground robot. Therefore, we address the problem of registering the maps computed by two robots from distant vantage points, using different sensing modalities: a dense 3D reconstruction from the MAV is aligned with the map computed from the depth sensor on the ground robot.

The most exciting opportunity of computer vision for mobile robotics is that robots can exhibit control on the data acquisition process. This motivated the investigation of the following problem: given the image of a scene, what is the trajectory that an MAV-mounted camera should follow to perform optimal dense depth estimation? The last contribution of this thesis addresses this question and introduces a method to compute the measurement uncertainty and, thus, the expected information gain, on the basis of the scene structure and appearance. This results in the MAV to choose motion trajectories that avoid perceptual ambiguities inferred by the texture in the scene.

Zusammenfassung

Kameras sind sehr nützliche Sensoren für mobile Roboter, da sie sehr klein, günstig und allgegenwertig sind. Weil jedes einzelne Kamerabild aus hunderttausenden Pixel-Messungen besteht, ist es eine grosse Herausforderung, aus dieser Datenflut die Kamerabewegung und gleichzeitig die Umgebung dreidimensional zu rekonstruieren. Noch schwieriger wird es, wenn dies in Echtzeit auf einer Recheneinheit mit beschränkter Kapazität, wie sie in Robotern eingesetzt wird, geschehen soll. Ausserdem wird die Robustheit des Systems ein sehr wichtiger Faktor, wenn der mobile Roboter sich in einer unkontrollierten Umgebung bewegt. In diesem Fall treten Verdeckungen, Beleuchtungsänderungen und wenig texturierte Oberflächen auf, was das Wiedererkennen von visuellen Merkmalen im Bild verhindert und deshalb die kamerabasierte Bewegungsschätzung erschwert.

Der erste Beitrag dieser Dissertation ist ein effizienter, robuster und sehr genauer Algorithmus für die visuelle Odometrie. Dieser Algorithmus schätzt die Bewegung einer einzelnen Kamera ausschliesslich anhand der von der Kamera aufgenommenen Bildern. Dazu wurden *direkte Methoden*, welche mit den Intensitätswerten der Pixel operieren, untersucht. Der Vorteil von direkten Methoden ist, dass die Pixel-Korrespondenz von Bild zu Bild durch die Geometrie des Problems gegeben ist und durch die Minimierung von Pixel-Intensitätsunterschieden weiter optimiert werden kann. Die Optimierung der Kamerapositionen und der 3D Geometrie der Umgebungsstruktur wird jedoch sehr rechenintensiv, wenn die Karte wächst. Daher wird ein halb-direkter (*semi-direct*) Algorithmus vorgeschlagen, der die Pixel-Korrespondenz mittels direkten Ansätzen ermittelt und daraufhin auf bewährten merkmal-basierten Methoden aufbaut, um die Geometrie des Problems zu optimieren. In einer Erweiterung wird gezeigt, wie Inertialmessungen nahtlos in diese Optimierierung integriert werden können. Dies stellt den zweiten Beitrag dieser Dissertation dar. Experimentelle Resultate zeigen, dass das vorgeschlagene System sehr genaue Schätzungen in Echtzeit erzielt, wobei insbesondere in Bezug auf die Rechenzeit signifikant bessere Resultate als im aktuellen Stand der Technik erreicht werden.

Das Rekonstruieren von visuellen Merkmalen in Video-Bildern resultiert in dünn besetzten Punktwolken. Ein Roboter hingegen braucht für die Manipulation, die Bewegungsplanung oder für das Ausweichen von Hindernissen eine dichte Representation

der Oberfläche. Das Ziel von früheren Arbeiten im Bereich der dichten Rekonstruktion von Oberflächen mittels Bildern ist meistens das Erzielen von möglichst hoher Genauigkeit. In der Robotik ist es hingegen sehr wichtig, dass man auch ein Mass für die Unsicherheit der Rekonstruktion schätzt, was als Mass für das Risiko bei der Bewegungsplanung oder für das optimale Fusionieren mit anderen Sensormodalitäten benutzt werden kann. Aus diesen Überlegungen entstand der dritte Beitrag dieser Dissertation: Ein Echtzeit Algorithmus für die probabilistische Rekonstruktion der Umgebung mittels einer einzelnen Kamera.
Eine spannende Anwendung von Computervision in der Robotik ist die Tatsache, dass der Roboter die Datenaufnahme beeinflussen kann. Daraus resultiert folgende Frage: Gegeben ist ein Bild der Umgebung; was ist die optimale Trajektorie, welche eine Kamera durchlaufen muss, um möglichst schnell die Tiefe jedes Pixels im Referenzbild zu schätzen? Diese Frage wird im letzten Beitrag dieser Dissertation untersucht. Dazu wird eine Methode vorgeschlagen, um die Messungenauigkeit und dadurch den Informationsgewinn jeder Kameraposition aufgrund der Struktur und Textur der Umgebung zu berechnen. Dies führt in Experimenten dazu, dass Roboter Trajektorien wählen, welche bildliche Doppeldeutigkeiten auflösen.

Contents

Abbreviations and Acronyms

2D	2-Dimensional
3D	3-Dimensional
DoF	Degrees of Freedom
EKF	Extended Kalman Filter
FAST	Features from Accelerated Segment Test
GPS	Global Positioning System
GPU	Graphics Processing Unit
IMU	Inertial Measurement Unit
MAV	Micro Aerial Vehicle
NEES	Normalized Estimation Error Squared
RMSE	Root Mean Squared Error
RPG	Robotics and Perception Group
ROS	Robot Operating System
SDF	Signed Distance Function
SfM	Structure-from-Motion
SLAM	Simultaneous Localization And Mapping
UAV	Unmanned Aerial Vehicle
VIO	Visual-Inertial Odometry
VO	Visual Odometry

1 Introduction

> AI has by now succeeded in doing essentially everything that requires 'thinking' but has failed to do most of what people and animals do 'without thinking.'
>
> *Donald Knuth* [Nilsson, 2009, p.318]

This thesis presents computer vision algorithms for motion estimation and mapping with autonomous mobile robots. While the algorithms presented in this thesis are general and can be employed in a wide variety of applications, the robotic demonstrator platforms used throughout this thesis are *Micro Aerial Vehicles (MAVs)*. MAVs are ideal platforms for a wide range of applications due to their small size and their unique capability to move in an agile way in three dimensional space. However, their small size limits the on-board processing capabilities and being unstable systems they require continuous and low-latency tracking of position and attitude. Furthermore, to autonomously perform useful applications they need to be fully aware of their three dimensional surroundings. In this respect, they present a more interesting and challenging application scenario than mobile robots that are constrained to move on a 2D surface.

The first part of this thesis investigates how MAVs can estimate their position and orientation in 3D space using only their on-board sensors and processors. Specifically, the focus lies on using only a single camera to achieve this goal and afterwards the fusion of cameras with inertial sensors is studied. Given the pose of the camera, the second part of the thesis investigates how a *dense* surface model can be efficiently obtained from the on-board images, which will allow the MAV to interact with the environment. This interaction is demonstrated in three application scenarios: autonomous landing, collaboration with ground-robots, and active dense reconstruction.

This thesis is structured in the form of a collection of papers. An introductory section that highlights the concepts and ideas behind the thesis is followed by seven self-

contained publications in the appendix.

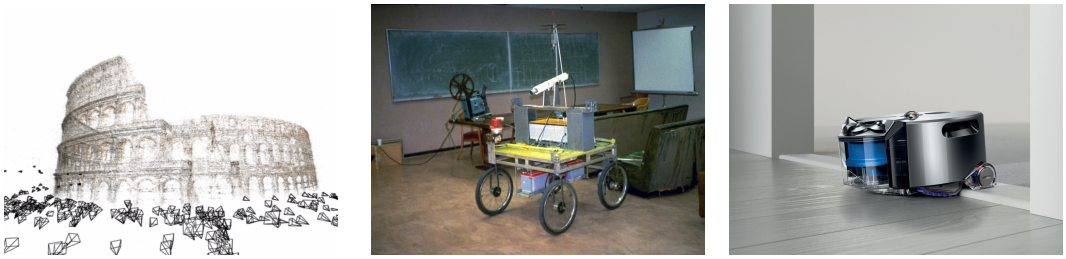
The next section introduces the general problem of vision-based robot perception and Section 1.2 highlights relevant related work in the field of MAV state estimation. Section 1.3 motivates and states the research objectives of this dissertation. Subsequently, Chapter 2 summarizes the key contributions of the papers in the appendix and explains the relationship among them. Finally, Chapter 3 suggests future research avenues of this work.

Cameras for Robot Perception

It seems effortless how humans perceive and interact with the environment. The ability and efficiency of the visual cortex to interpret the large amount of information perceived by the eyes is astonishing. Promptly we can describe our motion in the three dimensional space, characterize the size and structure of the room that we are in, and not easily are we deceived by reflections, shadows, and occlusions in the scene. Replicating human-scale understanding of space and motion in artificial systems represents an enormous challenge for scientists and engineers; however, even the smallest steps in this direction have the potential to unlock a myriad of exciting applications such as autonomous cars, service robots, or assistive devices for the blind. And indeed we have reasons to be optimistic: very rapid progress has been made in the last decade and continues to be made, aided by steady improvements in computing and sensing hardware. Today, computers are better than humans in detecting traffic signs in images [Stallkamp et al., 2012] and Google's autonomous car has autonomously driven more than a million miles [Google, 2015].

For an autonomous mobile robot to move around and share the space with humans, it needs to have a representation of the environment. The ideal representation depends on the scope of the robot's task. If the task of the robot is navigation, for instance to deliver a package, a useful environment representation is a map of landmarks that the robot is capable to detect with its on-board sensors. If this map is not existent a priori, the robot has to incrementally create a consistent map of landmarks while simultaneously determining its location within this map. This problem of *simultaneous localization and mapping* (SLAM) has been a cornerstone in robotics research. With the advent of probabilistic approaches in robotics, the problem could be formulated and solved in various forms [Durrant-Whyte and Bailey, 2006]. Today, SLAM algorithms are a standard module on mobile robots in research labs, which have been implemented using different sensor modalities, such as laser range finders, sonars, depth-cameras, or standard cameras.

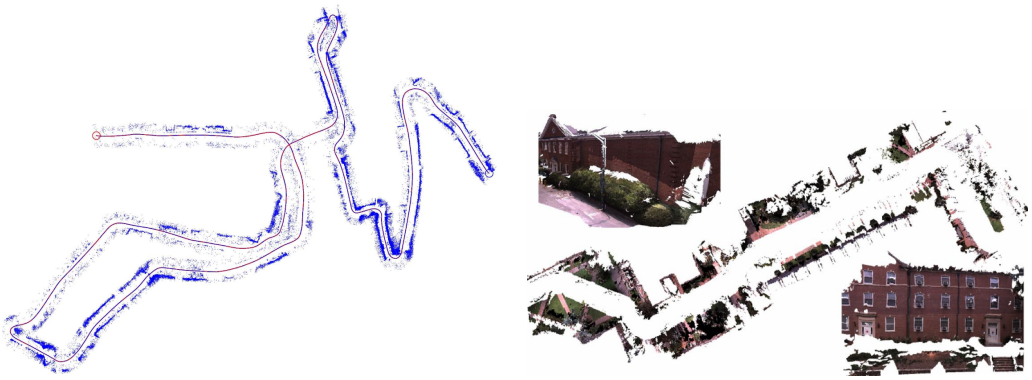
Using cameras for localization and mapping is appealing as these sensors are small, inexpensive, power efficient, and ubiquitous. Today, cameras are capable to record many



(a) (b) (c)

Figure 1.1 – (a) Reconstruction of the Colosseum in Rome from a large and unorganized collection of photographs taken from the internet using *structure from motion* [Agarwal et al., 2011]. **(b)** The Stanford Cart in 1970 by Moravec was the first robot to use a camera to estimate its ego-motion using *visual odometry* [Moravec, 1980]. **(c)** The Dyson 360 Eye^{TM} robotic vacuum cleaner is the first mass-market consumer product to use *visual-SLAM* technology. An omni-directional camera mounted at the top of the robot allows the robot to localize and navigate in homes.

times a second an image that consists of millions of individual pixel measurements. Interpreting this wealth of data is hard since the image formation process is affected by nuisance factors such as measurement noise, illumination changes, occlusions, quantization, among others, which affect the measurements but are irrelevant to the task at hand. The problem of reconstructing a 3D map of the environment from a set of possibly unordered images is called *structure from motion* (SfM) in the computer vision community. The origin of the approaches that address the structure and motion recovery from two images date back to the mid 1980s [Longuet-Higgins, 1981, Huang and Faugeras, 1987]. Today, it is possible to reconstruct large-scale models of buildings from unorganized set of hundred thousand photographs collected from the internet [Frahm et al., 2010, Agarwal et al., 2011] (see Fig. 1.1(a)). If the images appear in sequence, *e.g.*, from a camera mounted on a vehicle, we denote the incremental reconstruction of structure and motion as *visual odometry* (VO). Pioneered by Moravec [1980] in the 1970s (see Fig. 1.1(b)), the initial work on VO was mainly driven by NASA's Mars exploration program, which eventually led to the successful deployment on the Spirit and Opportunity rovers [Maimone et al., 2007]. In VO, the main goal is to minimize the drift in the motion estimate of the moving camera. However, if the goal is to recover a globally consistent map of the environment, *e.g.* by detecting previously places and subsequent refinement of the trajectory (loop closures), we speak of *visual SLAM* (although definitions may differ). The first systems that were capable to perform visual SLAM in *real-time* used an *Extended Kalman Filter* (EKF) to estimate structure and motion causally over time [Chiuso et al., 2002, Davison, 2003]. Today, the first mass marked consumer products start to use visual SLAM technology. A prominent example is the Dyson 360 Eye^{TM} robotic vacuum cleaner, which is equipped with an omni-directional camera that enables the robot to navigate and methodically clean rooms (see Fig. 1.1(c)).



(a) (b)

Figure 1.2 – Figure **(a)** depicts the recovered trajectory of a visual odometry (VO) algorithm. The algorithm proposed by Scaramuzza [2011] uses only a single omni-directional camera mounted on the roof of a car. The *sparse* point-cloud reconstruction of the static street-level environment is a by-product of the estimation process. In contrast, Figure **(b)** shows the *dense* reconstruction result by Pollefeys et al. [2008] that was computed in real-time from images recorded from a moving car.

At the heart of visual SLAM lies the correspondence problem: In order to estimate structure and motion from images, pixels that correspond to the same point in space need to be associated in successive images. The standard approach to this 2D – 2D image correspondence problem is to detect salient *features* in the images, compute a *descriptor* for each feature, and then match the descriptors between images. Corners are examples of salient features that can easily be identified as those pixels in the image that have a non-zero gradient along two independent directions [Harris and Stephens, 1988]. A descriptor such as the SIFT [Lowe, 2004] or BRISK [Leutenegger et al., 2011] is formed by computing the statistics of image gradients at different scales and locations in the feature neighborhood followed by various normalization and quantization. The goal of this procedure is to compute a descriptor that is invariant to illumination and view-point changes and thus can be reliably matched across images. Once feature correspondence is established, one can profit from the fruits of decades of geometric vision research that came up with minimal solutions to the inter-frame relative camera pose problems [Faugeras, 1994, Hartley and Zisserman, 2004, Ma et al., 2005]. Embedded in a recursive Bayesian estimation process, this allows us to recover the camera trajectory and a 3D map of landmarks. Due to the sparse nature of feature correspondence, the resulting map is also sparse (see Fig. 1.2(a)). Often just a by-product of motion estimation, the sparse point-cloud of 3D landmarks is a useful map representation to localize a robot. However, for more advanced robotic tasks a *sparse* map representation falls short. For instance, any sort of robot motion planning requires a *dense* surface representation to predict collisions and to enable true interaction with objects. Apart from active sensors such as laser range finders, a dense surface representation can also be computed from passive cameras using using dense

methods that exploit the information residing in all pixels of the images. Pollefeys et al. [1999, 2004] presented one of the earliest reconstruction pipelines that was capable to compute a dense surface model from a single camera. While the early work was running off-line on few images, later work in [Pollefeys et al., 2008] used a GPU to produce the first real-time capable dense reconstruction pipeline that was used to reconstruct street scenes from a car equipped with up to four non-overlapping cameras (see Fig. 1.2(b)).

The Advent of Micro Aerial Vehicles

In the last three years, we have heard a lot of news about micro aerial vehicles (MAVs): small flying robots less than a meter in size and typically equipped with four to eight rotors. While today MAVs are predominantly used as consumer toys, they are becoming more popular for personal photography and filming due to their unique capability of providing easy access to aerial perspectives. In the future, an even larger market will be the commercial use of MAVs for applications such as remote inspection of bridges, power plants, oil rigs, power lines, etc. (see Fig. 1.3(a)), agriculture (*e.g.*, for crop analysis), infrastructure (*e.g.*, management, surveillance), conservation, search and rescue, delivery (see Fig. 1.3(b)), cinematography, or journalism.

Today, MAVs are operated under direct line of sight by a trained operator. Successfully teleoperating an MAV is difficult and requires multiple days of training time. The reason is that multi-copters (*i.e.*, drones with multiple rotors) are inherently unstable systems that require continuous regulation of attitude and position.

Integrated solutions for commercial applications will have to maximize safety and minimize the risk of injury to humans. There will be the need for advanced autopilots that assist operators in difficult terrains (*e.g.*, close to buildings), out of line-of-sight operation, or for completely autonomous flight. To date, most autopilots of commercial MAVs rely on satellite-based global positioning systems (GPS). However, GPS is not reliable in urban settings and is completely unavailable indoors. Furthermore, GPS does not provide the situation awareness to allow obstacle avoidance, position lock, or safe emergency landing. Therefore, large-scale Unmanned Aerial Vehicles (UAVs) often use range sensors to detect hazards, avoid obstacles, or to land autonomously [Johnson et al., 2002, Scherer et al., 2012]. However, these sensors are expensive, heavy and quickly drain the battery when used on MAVs.

Thanks to advances in sensor and processor miniaturization for consumer electronics, research on MAVs has progressed significantly in the last years. The key problem in designing autopilots for MAV navigation is attitude and position control. By fusing inertial sensors (*i.e*, gyroscopes and accelerometers), the roll and pitch axis of the MAV can be reliably controlled while hovering. However, without any other exteroceptive

(a) (b)

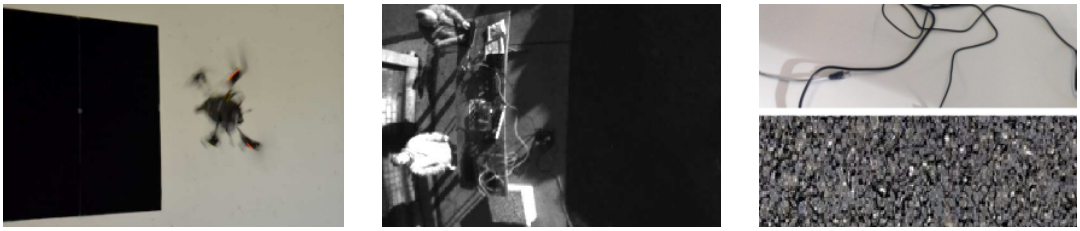
Figure 1.3 – (a) Sensefly's eXom inspection MAV for industrial inspection. **(b)** Amazon's delivery drone.

sensory input (*e.g.*, cameras, laser rangefinders), the MAV drifts quickly in position and yaw direction. 2D laser range-finders have been largely explored for ground mobile robots [Thrun et al., 2005] and similar strategies have been extended to MAVs [Achtelik et al., 2009, Shen et al., 2012]. Although laser scanners are very reliable and robust, today's sensors are too heavy and consume too much power for lightweight MAVs. Therefore, vision sensors are very appealing, their effective range is not limited as in the case of time-of-flight or camera-projector systems allowing to both very far and very close operation to the surface. However, being passive sensors, they require advanced computer vision algorithms that interpret in real-time the vast amount of pixel data provided by the cameras—which has become feasible for on-board computers only in the last decade through the rapid improvements in smart-phone processors.

Early works on vision-based MAV navigation have used biologically-inspired control algorithms, such as optical flow [Zufferey and Floreano, 2006, Zingg et al., 2010, Grabe et al., 2013]. However, since optical flow only measures the relative velocity, drift in the MAV position is still inevitable. Therefore, recent camera-based autopilots used advanced computer vision algorithms relying on visual odometry (VO) technology to perform autonomously basic maneuvers, such as take off, landing, and way-point–based navigation [Scaramuzza et al., 2014, Faessler et al., 2015]. VO estimates the camera trajectory and structure of the scene from images and relying on these algorithms completely eliminates drift in hover condition.

Goal and Motivation of this Dissertation

In this section I describe the open challenges in vision-based motion estimation and dense mapping with MAVs, which leads to a summarization of the proposed approaches in this thesis.



(a) High speeds. (b) Illumination changes. (c) Difficult texture.

Figure 1.4 – The dominant challenges for the successful application of VO on-board MAVs is robustness during agile and fast maneuvers, during illumination changes, and in scenes of difficult texture such as high-frequency or repetitive texture.

Robust, Accurate, and Efficient Visual Odometry

The challenges for the successful adoption of VO in commercial autopilots of MAVs are to increase *robustness* and to improve *accuracy* and *efficiency*.

Robustness For VO on-board MAVs it is crucial to guarantee robustness in case of fast motions, illumination changes, motion blur, low textured environments, or in the presence of dynamic obstacles as illustrated in Fig. 1.4. These nuisances increase the difficulty to track visual cues, which is fundamental to enable vision-based motion estimation. A VO system is most robust if selected pixels can be reliably tracked from frame to frame and the motion is such that the 3D position of those pixels may be triangulated accurately. The probability that many pixels are tracked reliably, *e.g.*, in scenes with little or high frequency texture (such as sand [Maimone et al., 2007] or asphalt [Lovegrove et al., 2011]), is increased when the algorithm is not restricted to use local point features (*e.g.*, corners or blobs) but may track edges [Klein and Murray, 2008] or more generally, all pixels with gradients in the image, such as in dense [Newcombe et al., 2011b] or semi-dense approaches [Engel et al., 2014]. Dense or semi-dense algorithms that operate directly on pixel-level intensities are also denoted as *direct methods* [Irani and Anandan, 1999]. Direct methods estimate structure and motion directly by minimizing an error measure that is based on image's pixel-level intensities. The local intensity gradient magnitude and direction is used in the optimization compared to feature-based methods that consider only the distance to a feature-location. Pixel correspondence is given directly by the geometry of the problem, eliminating the need for robust data association techniques. Direct methods also benefit from the use of high frame-rate cameras [Handa et al., 2012]: the underlying optimization problem converges faster and the capability to track weak gradients is further improved. Furthermore, the use of dense direct methods improves the robustness of the system against nuisances such as motion blur or camera defocus, as shown exemplary in [Newcombe et al., 2011b].

Accuracy In terms of accuracy, the optimal camera motion estimate is obtained through joint optimization of structure (*i.e.*, landmarks) and motion (*i.e.*, camera poses). For *sparse feature correspondence*, this is an established problem that is commonly known as *bundle adjustment* [Triggs et al., 2000] and many solvers exist that address the underlying non-linear least-squares problem efficiently [Dellaert and Kaess, 2006, Agarwal et al., Kümmerle et al., 2011]. However, joint optimization of *dense* structure and motion in real-time is still an open research problem. For this reason, the standard approach for dense VO is to estimate the latest camera pose with respect to a previously accumulated dense map and subsequently, given a set of estimated camera poses, update the dense map [Newcombe et al., 2011b, Ondruska et al., 2015]. Clearly, this separation of tracking and mapping is only optimal when the output of each stage yields the optimal estimate. Other algorithms optimize a graph of poses but do not allow a deformation of the structure once triangulated [Engel et al., 2014]. Contrarily, some algorithms ignore the camera poses and instead allow non-rigid deformation of the 3D structure [Whelan et al., 2014, 2015]. The obtained results are accurate and visually impressive, however, the algorithms lack a thorough probabilistic treatment when separating tracking and mapping or fixating and removing states. Therefore, existing dense approaches cannot guarantee optimal and *consistent* [Huang et al., 2010] fusion of visual and, if available, complementary measurements (*e.g.*, inertial). An estimator is said to be consistent if the estimation errors match the theoretical statistical characteristics, *i.e.*, they are zero mean (unbiased) and have a covariance as calculated by the estimator [Bar-Shalom et al., 2001].

Efficiency The third requirement for the successful use of VO on-board MAVs is efficiency. Differently from many state-of-the-art systems in computer vision, the robotic application on-board MAVs puts hard constrains on the update rate of the VO. Ideally, the use of very small MAVs is desired for safety reasons. However, the small size comes with low inertia and thus higher agility. These fast dynamics of the flying robot require a high update rate of the VO. Additionally, small MAVs are typically equipped with computationally constrained processors and a limited power budget. Since every camera image provides hundred thousands of measurements, it poses a great challenge to infer motion from this wealth of data in real-time on computationally constrained platforms. Not surprisingly, the primary performance limitation for VO on the Mars rovers was the runtime of the image processing part. This was identified by Maimone et al. [2007] as the overriding immediate priority for future flight computers. Although meanwhile processors have significantly improved, more efficient VO algorithms are still very desirable to minimize both the power consumption and the required computational budget on-board MAVs.

The discussion on robustness, accuracy, and efficiency highlights that for increased robustness, it is beneficial to use dense direct methods that exploit information from all gradients in the image; conversely, for highest accuracy and fusion with complementary

sensors, it is advantageous to rely on proven feature-based methods that are efficient and guarantee optimality and consistency. Finally, for efficiency, both dense methods, which process every pixel in an image, as well as feature-based methods, which require expensive computation of features and descriptors for every image, are a disadvantage.

Therefore, in [Forster et al., 2014b], we proposed a VO pipeline that combines the advantages of direct and feature-based methods: we rely on direct methods to robustly establish feature correspondence and once matches are established, we use bundle adjustment for refinement of the 3D structure and the camera poses. Consequently, the system is called *semi-direct* visual odometry (SVO). The approach is very efficient since feature extraction is not necessary for every frame and for direct alignment only pixels in the feature neighborhood are considered. Our implementation is very modular, which allowed us to extend it to multi-camera systems and wide field of view lenses. Additionally, we proposed a method to use inertial measurements from an inertial measurement unit (IMU) in the bundle adjustment problem. It has been shown that adding inertial measurements to the bundle adjustment problem results in highly accurate state estimation [Jung and Taylor, 2001, Sterlow and Singh, 2004, Indelman et al., 2013b]. However, real-time optimization quickly becomes infeasible as the trajectory grows over time; this problem is further emphasized by the fact that inertial measurements come at high rate, hence leading to fast growth of the number of variables in the optimization. This issue can be addressed by preintegrating inertial measurements between selected keyframes into single relative motion constraints, which was first proposed by Lupton and Sukkarieh [2012]. We build upon this work and present a preintegration theory that properly addresses the manifold structure of the rotation group $\mathrm{SO}(3)$. Compared with [Lupton and Sukkarieh, 2012], our theory offers a more formal treatment of the rotation noise, and avoids singularities in the representation of rotations.

Dense Reconstruction for Mobile Robots

The typical map representation that is recovered by a VO algorithm is a *sparse* point-cloud of which every 3D point is triangulated from salient features tracked in the video stream (*e.g.*, see Figure 1.2(a)). However, for robotic tasks such as path planning, manipulation, or obstacle avoidance, a *dense* reconstruction (*e.g.*, see Figure 1.2(b)) is needed to interact with the environment.

There exist a myriad of works on dense reconstruction from images in the computer vision literature. I refer to well known benchmarks in [Seitz et al., 2006] and [Strecha et al., 2008] for a representative lists. The input of these algorithms is a set of calibrated images with known pose as it can be computed by an accurate visual SLAM algorithm. The primary goal of these benchmarks is to create high fidelity reconstructions such as in [Zach et al., 2007, Goldlücke et al., 2009, Furukawa and Ponce, 2010, Tola et al.,

2012, Fuhrmann and Goesele, 2014]. However, highest accuracy and resolution is not necessarily the first priority when dense reconstruction is used in mobile robotics. In this case, energy and processing time constraints apply and the level of reconstruction detail should be governed by the interaction task of the robot. An MAV for instance does not require a surface reconstruction with millimeter precision to select a suitable landing spot. On the other hand, highest surface density may be required by a mobile manipulator for grasping an object. Robotics further requires surface representations that enable the robot to reason more efficiently about future actions. Therefore, a measure of uncertainty in the 3D reconstruction is necessary. However, in many dense reconstruction pipelines regularization techniques are used that hallucinate data for instance by interpolating the surface in regions without texture. A robot must be aware of the uncertainty of these estimates and take it into consideration for planning. A probabilistic depth representation is further a prerequisite for optimal fusion with different sensing modalities.

Probabilistic Dense Reconstruction with a Single Camera

Dense depth estimation from a single moving camera is an appealing sensing modality for MAVs, where strict limitations on payload and power consumption apply. In this case, the high agility turns the platform into a formidable depth sensor, able to deal with a large depth range and capable of achieving arbitrarily high confidence in the measurement.

Few relevant works have addressed real-time, dense reconstruction from a single moving camera and they shed light on some important aspects. If, on one hand, estimating the depth independently for every pixel leads to efficient, parallel implementations, on the other hand the authors of [Gallup et al., 2007, Stühmer et al., 2010, Newcombe et al., 2011b] argued that, similar to other computer vision problems, such as image de-noising [Rudin et al., 1992] and optical flow estimation [Werlberger et al., 2010], a smoothing step is required in order to deal with noise and spurious measurements. In [Stühmer et al., 2010], smoothness priors were enforced over the reconstructed scene by minimizing a regularized energy functional based on aggregating a photometric cost over different depth hypothesis and penalizing non-smooth surfaces. [Newcombe et al., 2011b, Vogiatzis and Hernández, 2011] further showed that the integration of many images leads to significantly higher robustness to noise. However, despite the ground-breaking results, these approaches present some limitations when addressing tasks in robot perception. Equally weighting measurements from small and large baselines, in close and far scenes, causes the aggregated cost to frequently present multiple or no minima. Depending on the depth range and sampling, these failures are not always recoverable by the subsequent optimization step. Furthermore, an inadequate number of images can lead to a poorly constrained initialization for the optimization and erroneous measurements that are hard to detect. It is not clear how

many images should be collected, depending on the motion of the camera and the scene structure.

In our work [Pizzoli et al., 2014], we rely on the camera pose estimation of the SVO odometry system [Forster et al., 2014b] and build upon the work of Vogiatzis and Hernández [2011] for per-pixel recursive Bayesian depth estimation from a single camera. This results in a *probabilistic* representation of the depth that reflects our confidence in the reconstruction and allows the robot to reason about future motions that minimize the uncertainty [Forster et al., 2014a]. We further propose an optimization step to enforce spatial regularity over the recovered depth map. The novelty of the regularization is that the estimated depth uncertainty from the per-pixel depth estimation is used to weight the smoothing.

Dense Reconstruction applied to MAVs

While most computer vision approaches fuse depth-maps in a cost volume, care has to be taken in terms of scalability and memory consumption for MAV applications. Therefore, we propose a robot-centric elevation map of fixed size that suits a flying robot with down-looking camera [Forster et al., 2015c]. The map representation is memory efficient as it can be implemented with a two dimensional rolling buffer. In experiments we have demonstrated that this map representation is effective for emergency landing spot detection.

We further demonstrate the usefulness of dense depth-maps for localization of an MAV with respect to a ground robot [Forster et al., 2013]. Therefore, we solve the problem of registering the maps computed by the robots using different sensors: a dense 3D reconstruction from the MAV is aligned with the map computed from the depth sensor on the ground robot. Once aligned, the dense reconstruction from the MAV is used to augment the map computed by the ground robot, by extending it with the information conveyed by the aerial views.

Active Dense Reconstruction

The most exciting opportunity of computer vision for robotics is that robots can exhibit control on the data acquisition process. Most related work in computer vision is *passive* in a sense that algorithms process datasets that were recorded in a previous time instant. In contrast, in robotic applications, we can develop algorithms that are *active* by executing some authority on the choice of motion of the MAV or the camera acquisition parameters (*e.g.*, shutter speed, frame-rate). This motivated us to investigate the following problem: given the image of a scene, what is the trajectory that a robot-mounted camera should follow to perform optimal dense depth estimation? State-of-the-art approaches to active mapping [Kriegel et al., 2011, Bourgault et al.,

2002, Davison and Murray, 2002, Stachniss et al., 2005, Valencia et al., 2012, Sim and Roy, 2005] retain only geometric information while discarding the scene appearance. As a result, a robot trying to perceive the depth of a white wall, would generate different camera trajectories in vain, eventually failing to reduce the uncertainty in the depth measurement [Soatto, 2009]. By contrast, our proposed method computes the measurement uncertainty and, thus, the expected information gain, on the basis of scene structure *and* appearance (*i.e.*, texture) [Forster et al., 2014a]. To the best of our knowledge, this is the first work on *active, monocular dense* reconstruction, which chooses motion trajectories that minimize perceptual ambiguities inferred by the texture in the scene.

2 Contributions

This chapter summarizes the key contributions of the papers that are reprinted in the appendix. It further highlights the connections between the individual results and refers to related work and video contributions. In total, this research been published in 9 peer-reviewed conference publications and two journal publication. The work [Forster et al., 2015b] on visual-inertial estimation was best paper award finalist at *Robotics: Science and Systems 2015*. Two additional papers have been conditionally accepted for publication in IEEE Transactions on Robotics.

Robust, Accurate, and Efficient Visual Odometry

Paper A: Feature-based Multi-Robot Visual SLAM

(P1) C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza. Collaborative monocular SLAM with multiple micro aerial vehicles. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3962–3970, 2013. URL http://dx.doi.org/10.1109/IROS.2013.6696923.

In this work a framework for collaborative visual SLAM with multiple MAVs is presented. Each MAV estimates its motion individually using a monocular visual odometry algorithm that runs on-board. The MAVs collectively act as distributed preprocessors that stream only features and descriptors of selected *keyframes* and relative-pose estimates to a centralized ground station. The ground station creates an individual map for each MAV and merges them whenever overlaps are detected. After map merging, the position of the MAVs can be expressed in a common, global coordinate frame. The key to real-time performance is the design of data-structures and processes that allow multiple threads to concurrently read and modify shared global map. The presented framework is tested in both indoor and outdoor environments with up to three MAVs. To the best of our knowledge, this is the first work on real-time collaborative monocular SLAM, which has also been applied to MAVs.

Related Videos

(V1) C. Forster, S. Lynen, L. Kneip and D. Scaramuzza (2012): "Collaborative Monocular SLAM with Multiple Micro Aerial Vehicles." https://youtu.be/taD3XF2w7A0.

Paper B: Semi-Direct Visual Odometry (SVO)

(P2) C. Forster, Z. Zhang, M. Gassner, M. Werlberger, D. Scaramuzza. Semi-Direct Visual Odometry for Monocular and Multi-Camera Systems. *IEEE Transactions on Robotics (TRO)*, 2016 (accepted).
PDF: rpg.ifi.uzh.ch/docs/TRO16_forster_SVO.pdf

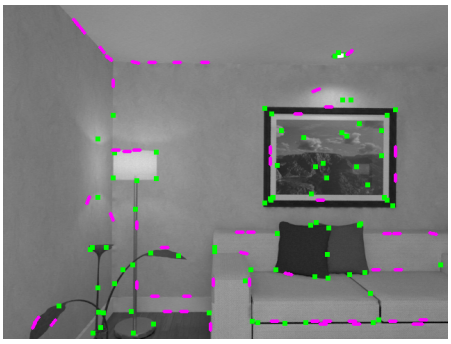
The computation of features and descriptors for every frame in the previous work proved computationally to be very intensive. Therefore, computation on-board MAVs was limited to low frame-rates, which in turn limited the agility of the vehicle. This motivated the development of a more robust and efficient VO system.

With "SVO", we proposed a VO pipeline that combines the advantages of direct and feature-based methods. Direct methods for VO that operate directly on pixel level intensities [Irani and Anandan, 1999] have recently gained popularity due to their capability to exploit information from all image gradients in the image. However, low computational speed in large-scale problems as well as missing guarantees for optimality and consistency are limiting factors of direct methods where established feature-based methods instead succeed at. Based on these considerations, we proposed a semi-direct approach that uses direct methods to track and triangulate features but relies on proven feature-based methods (bundle-adjustment [Triggs et al., 2000]) for refinement and fusion with additional sensors. The main novelty is the sparse-image-alignment algorithm that tracks a set of features with known scene depth jointly from frame to frame satisfying epipolar constrains. In conjunction with a direct and robust depth estimation algorithm, this approach allows tracking of weak corner features and edgelets in environments with little or high-frequency texture (see Fig. 2.1). The proposed algorithm is very flexible and can easily be extended to multiple cameras, use motion priors, and wide field of view lenses. Experimental evaluation on benchmark datasets shows that the algorithm is significantly faster than the state of the art while achieving highly competitive accuracy.

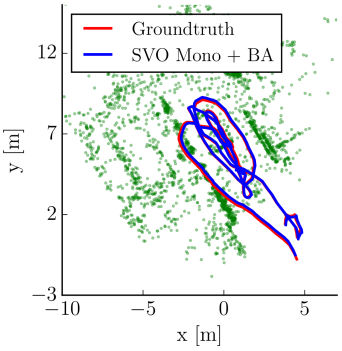
As a central component of the auto-pilot developed at the Robotics and Perception Group (RPG), SVO has been used in the last years on a daily basis for experiments with MAVs. Our MAV system was demonstrated over 300 times at the RPG lab, at multiple trade fairs, at public events, and to search and rescue professionals. Our implementation is further used for several commercial products such as for state-estimation on-board MAVs and for tracking the pose of a smart-phone for reconstruction applications. Furthermore, other research groups, such as the Autonomy and Robotics Center at NASA Langley, are successfully using the open-source SVO algorithm for MAV applications.

Related Publications

(R1) C. Forster, M. Pizzoli, and D. Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 15–22, 2014. URL http://dx.doi.org/10.1109/ICRA.2014.6906584.



(a) Tracking performance of SVO on the ICL-NUIM dataset [Handa et al., 2014]. SVO uses both corner (green) and edgelet (magenta) features.



(b) Estimated trajectory and point-cloud on the Euroc dataset [Burri et al., 2015], which was recorded with an MAV in a machine hall.

Figure 2.1 – Results of the SVO algorithm.

(R2) M. Faessler, F. Fontana, C. Forster, E. Mueggler, M. Pizzoli, and D. Scaramuzza. Autonomous, vision-based flight and live dense 3D mapping with a quadrotor MAV. In *Journal of Field Robotics*, pages 1556–4967, 2015. URL http://dx.doi.org/10.1002/rob.21581.

(R3) Z. Zhang, H. Rebecq, C. Forster, and D. Scaramuzza. Benefit of large field-of-view cameras for visual odometry. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2016. URL http://dx.doi.org/10.1109/ICRA.2016.7487210.

(R4) M. Faessler, F. Fontana, C. Forster, and D. Scaramuzza. Automatic re-initialization and failure recovery for aggressive flight with a monocular vision-based quadrotor. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1722–1729, 2015. URL http://dx.doi.org/10.1109/ICRA.2015.7139420.

Related Demonstrations

(D1) C. Forster, M. Pizzoli, and D. Scaramuzza. SVO: Fast semi-direct monocular visual odometry. Live demonstration at *Eur. Conf. on Computer Vision (ECCV)*, Zurich, 2014.

Related Software

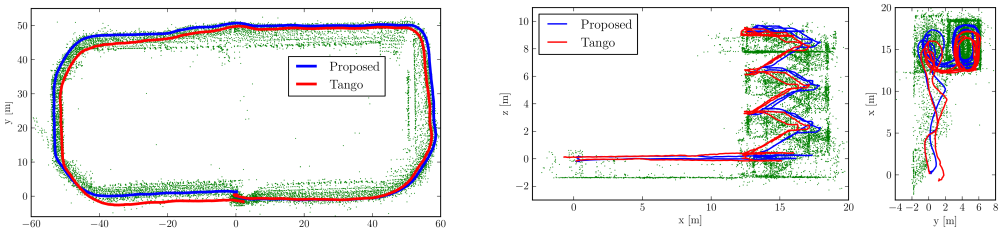
(S1) https://github.com/uzh-rpg/rpg_svo.

Related Videos

(V2) C. Forster, M. Pizzoli, and D. Scaramuzza (2014): "SVO: Fast Semi-Direct Monocular Visual Odometry." https://youtu.be/2YnIMfw6bJY.

(V3) M. Faessler, F. Fontana, C. Forster, and D. Scaramuzza (2015): "Automatic Re-Initialization and Failure Recovery for Aggressive Flight with a Monocular Vision-Based Quadrotor." https://youtu.be/pGU1s6Y55JI.

(V4) C. Forster, Z. Zhang, M. Gassner, M. Werlberger, D. Scaramuzza (2015): "Semi-Direct Visual Odometry for Monocular and Multi-Camera Systems." https://youtu.be/hR8uq1RTUfA.



(a) (b)

Figure 2.2 – Comparison of the proposed visual-inertial odometry system in [Forster et al., 2015b] against the *Google Tango* sensor. In (a) the Tango sensor accumulated 2.2m drift while the proposed estimator achieved 1.0m error at the end of the trajectory. In (b) the proposed approach exhibits 0.5m drift while the Tango sensor accumulated 1.4m at the end of the trajectory.

Paper C: Visual-Inertial Odometry

(P3) C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. On-Manifold Preintegration for Real-Time Visual-Inertial Odometry. *IEEE Transactions on Robotics (TRO)*, 2016. URL: http://rpg.ifi.uzh.ch/docs/TRO16_forster_VIO.pdf.

The use of an inertial measurement unit (IMU) as a complementary sensor to the camera promises increased robustness and accuracy for VO. While a single moving camera is an exteroceptive sensor that allows us to measure appearance and geometry of a three-dimensional scene, up to an unknown metric scale; an inertial measurement unit (IMU) is a proprioceptive sensor that renders metric scale of monocular vision and gravity observable [Martinelli, 2013] and provides robust and accurate inter-frame motion estimates. This motivated the development of a visual-inertial odometry system.

Current approaches for visual-inertial odometry are able to attain highly accurate state estimation via nonlinear optimization. However, real-time optimization quickly becomes infeasible as the trajectory grows over time; this problem is further emphasized by the fact that inertial measurements come at high rate, hence leading to fast growth of the number of variables in the optimization. This issue can be addressed by preintegrating inertial measurements between selected keyframes into single relative motion constraints [Lupton and Sukkarieh, 2012]. We build upon this work and propose a *preintegration theory* that properly addresses the manifold structure of the rotation group. We formally discuss the generative measurement model as well as the nature of the rotation noise, which leads to the derivation of the expressions for the maximum a posteriori state estimator. This theoretical development enables the computation of all necessary Jacobians for the optimization and a-posteriori bias correction in analytic form. The second contribution is to show that the preintegrated IMU model can be seamlessly integrated into a visual-inertial pipeline under the unifying framework of factor graphs. This enables the application of incremental-smoothing algorithms and

the use of a *structureless* model for visual measurements, which avoids optimizing over the 3D points, further accelerating the computation. We perform an extensive evaluation of our monocular visual-inertial pipeline on real and simulated datasets. The results confirm that our modeling effort leads to accurate state estimation in real-time, outperforming state-of-the-art approaches (see Fig. 2.2(b)).

Related Publications

(R5) C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. In *Robotics: Science and Systems (RSS)*, 2015. URL https://dx.doi.org/10.15607/RSS.2015.XI.006.

(R6) C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. Supplementary Material to: IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. *Technical Report GT-IRIM-CP&R-2015-001*, 2015. URL http://hdl.handle.net/1853/53653.

Related Software

(S2) https://bitbucket.org/gtborg/gtsam.

Related Videos

(V5) C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza (2015): "IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation." https://youtu.be/CsJkci5lfco.

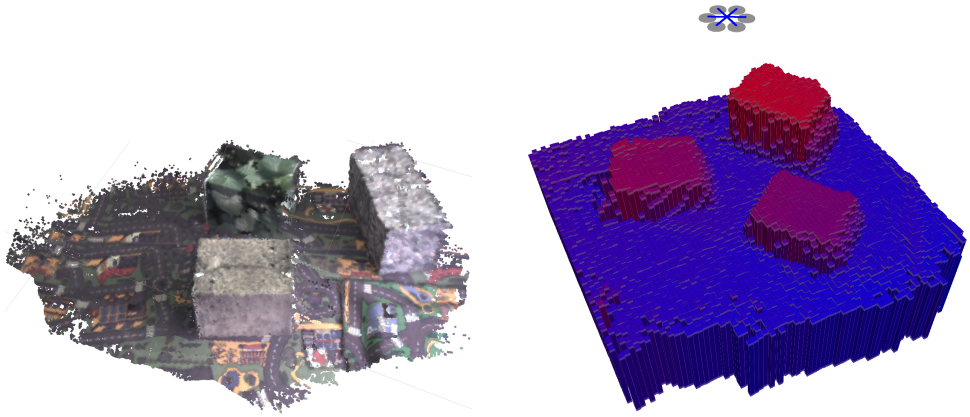
Dense Reconstruction

Paper D: Probabilistic Depth-Map Estimation

(P4) M. Pizzoli, C. Forster, and D. Scaramuzza. REMODE: Probabilistic, monocular dense reconstruction in real time. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2609–2616, 2014. URL http://dx.doi.org/10.1109/ICRA.2014.6907233.

The map representation recovered by the VO algorithms presented in the previous section is a sparse point-cloud of which every 3D point is triangulated from salient features tracked in the video stream. However, for robotic tasks such as path planning, manipulation, or obstacle avoidance, a *dense* reconstruction is needed to interact with the environment (see Fig. 2.3(a)). This motivated the development of a real-time capable vision-based dense reconstruction pipeline.

Our main contribution in this field is a novel approach to depth-map computation that combines Bayesian estimation and recent developments in convex optimization for image processing. We estimate the camera trajectory with the VO algorithm proposed in [Forster et al., 2014b] and compute probabilistic depth maps with the recursive Bayesian scheme from [Vogiatzis and Hernández, 2011]. Therefore, a per-pixel depth estimation is carried out for selected reference frames. We further propose a fast



(a) Reconstruction results [Pizzoli et al., 2014] computed real-time but off-board.

(b) Elevation map [Forster et al., 2015c] computed in real-time on-board the MAV.

Figure 2.3 – Dense reconstruction for MAVs.

smoothing method that takes into account the estimation uncertainty to provide spatial regularity and mitigate the effect of noisy camera localization. We demonstrate that our method outperforms the state-of-the-art in terms of accuracy, while exhibiting high efficiency in memory usage and computing power.

Related Software

(S3) https://github.com/uzh-rpg/rpg_open_remode.

Related Videos

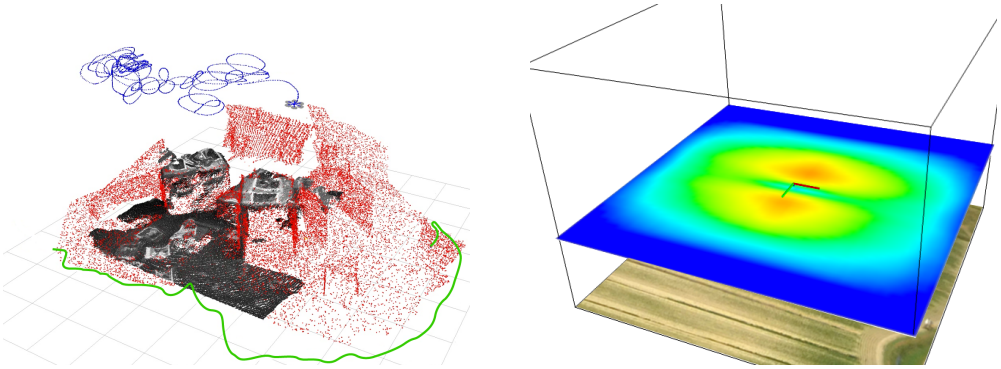
(V6) M. Pizzoli, C. Forster, and D. Scaramuzza (2014): "REMODE: Probabilistic, monocular dense reconstruction in real time." https://youtu.be/QTKd5UWCG0Q.

(V7) M. Faessler, F. Fontana, C. Forster, E. Mueggler, M. Pizzoli, D. Scaramuzza (2015): "Autonomous, Flying 3D Scanner." https://youtu.be/7-kPiWaFYAc.

Paper G: Autonomous Landing Using Dense Reconstruction

(P5) C. Forster, M. Faessler, F. Fontana, M. Werlberger, and D. Scaramuzza. Continuous on-board monocular-vision-based aerial elevation mapping for quadrotor landing. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 111–118, 2015. URL http://dx.doi.org/10.1109/ICRA.2015.7138988.

Our previous work in [Pizzoli et al., 2014] computes dense probabilistic depth maps in real-time but off-board on a graphics processing unit (GPU). However, for increased autonomy, the MAV cannot rely on a connection to a base station. This motivated the development of a reconstruction pipeline that can run on-board the MAV and increases the autonomy of the robot.



(a) Collaboration of an aerial and ground robot. The aerial robot augments the reconstructions from the ground-robot with a dense reconstruction obtained from aerial views.

(b) Information gain predicted for the next measurement. A motion orthogonal to the striped surface texture is optimal [Forster et al., 2014a].

Figure 2.4 – Application of dense reconstruction for MAVs.

In this work, we propose a resource-efficient system for real-time 3D terrain reconstruction and landing-spot detection for MAVs. The system runs on an on-board smart-phone processor and requires only the input of a single down-looking camera and an inertial measurement unit. We generate a two-dimensional elevation map that is probabilistic, of fixed size, and robot-centric, thus, always covering the area immediately underneath the robot (see Fig. 2.3(b)). The elevation map is continuously updated at a rate of 1 Hz with depth maps that are triangulated from multiple views using recursive Bayesian estimation. To highlight the usefulness of the proposed mapping framework for autonomous navigation of MAVs, we successfully demonstrate fully autonomous landing including landing-spot detection in real-world experiments.

Related Videos

(V8) C. Forster, M. Faessler, F. Fontana, M. Werlberger, and D. Scaramuzza (2015): "Continuous on-board monocular-vision–based aerial elevation mapping for quadrotor landing." https://youtu.be/phaBKFwfcJ4.

Paper F: Air-Ground Matching using Dense Reconstruction

(P6) C. Forster, M. Pizzoli, and D. Scaramuzza. Air-ground localization and map augmentation using monocular dense reconstruction. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3971–3978, 2013. URL http://dx.doi.org/10.1109/IROS.2013.6696924.

In this work, we demonstrate in another application the usefulness of on-board dense reconstruction for MAVs. We address the problem of registering the 3D maps computed by the robots using different sensors: a dense 3D reconstruction from the MAV monocular camera is aligned with the map computed from the depth sensor on the

ground robot. Once aligned, the dense reconstruction from the MAV is used to augment the map computed by the ground robot, by extending it with the information conveyed by the aerial views. In spite of the radically different vantage points from which the maps are acquired, the proposed approach achieves high accuracy whereas appearance-based, state-of-the-art approaches fail. Experimental validation in indoor and outdoor scenarios reported an accuracy in position estimation of 0.08 meters and real time performance. This demonstrates that our new approach effectively overcomes the limitations imposed by the difference in sensors and vantage points that negatively affect previous techniques relying on matching visual features.

Related Videos

(V9) C. Forster, M. Pizzoli, and D. Scaramuzza (2013): "Air-ground localization and map augmentation using monocular dense reconstruction." https://youtu.be/IZJmZIbinGg.

Paper E: Active Dense Reconstruction

(P7) C. Forster, M. Pizzoli, and C. Scaramuzza. Appearance-based active, monocular, dense depth estimation for micro aerial vehicles. In *Robotics: Science and Systems (RSS)*, 2014. URL https://dx.doi.org/10.15607/RSS.2014.X.029.

In this work, we investigate the following problem: given the image of a scene, what is the trajectory that a robot-mounted camera should follow to allow optimal dense depth estimation? The solution we propose is based on maximizing the information gain over a set of candidate trajectories. State-of-the-art approaches to active mapping [Kriegel et al., 2011, Bourgault et al., 2002, Davison and Murray, 2002, Stachniss et al., 2005, Valencia et al., 2012, Sim and Roy, 2005] retain only geometric information while discarding the scene appearance. As a result, a robot trying to perceive the depth of a white wall, would generate different camera trajectories in vain, eventually failing to reduce the uncertainty in the depth measurement [Soatto, 2009]. By contrast, we proposed a method to compute the measurement uncertainty and, thus, the expected information gain, on the basis of scene structure *and* appearance (i.e., texture). For applications to dense reconstruction from MAVs, we provided a strategy to compute a candidate sequence of viewpoints that lie on a feasible trajectory and that maximize the expected information gain. We obtain both synthetic and experimental validation of the proposed system in closed loop and compare against four different control strategies: a random strategy, a circular motion, a greedy strategy and a Next-Best-View (NBV) strategy that iteratively selects the globally optimal view points. To the best of our knowledge, this is the first work on *active, monocular dense* reconstruction, which chooses motion trajectories that minimize perceptual ambiguities inferred by the texture in the scene (see Fig. 2.4(b)).

Related Publications

(R7) G. Costante, C. Forster, P. Valigi, D. Scaramuzza. Perception-aware Path Planning. *Submitted to IEEE Transactions on Robotics (TRO)*.

Related Videos

(V10) C. Forster, M. Pizzoli, and C. Scaramuzza (2014): "Appearance-based active, monocular, dense depth estimation for micro aerial vehicles." https://youtu.be/uAc1pL_c-zY.

3 Future Directions

Given the progress in visual odometry (VO) and dense reconstruction in recent years, the technology has reached a level of maturity that allows its use in commercial robotic systems. In consumer MAVs for instance, visual odometry will soon be used to support teleoperated flight in GPS-denied environments and to increase the hover stability. Vision-based dense reconstruction techniques will be used to detect obstacles and safe landing spots. However, to enable fully autonomous MAVs in commercial applications, robustness remains an issue. In particular, it will be difficult to certify robustness of vision-based algorithms. Therefore, significant engineering efforts are still required, which should become the task of the industry that is applying this technology. On the other hand, the work in this thesis can be continued along several exciting research avenues:

Novel Sensing Technologies There is a major discrepancy between the available camera technology today and the sensors commonly used for VO research. While high resolution and high frame-rate cameras are a commodity today, research on VO is still focused on low resolution monochrome cameras. For instance, there is a trade-off between using direct pixel tracking in high frame-rate cameras versus computing robust descriptors for wide-baseline data association in low frame-rate cameras. Moreover, there exist a variety of completely novel cameras such as event-based neuromorphic vision sensors [Delbruck et al., 2010, Mueggler et al., 2015]. These sensors output a low-latency stream of "events" that is generated when single pixels perceive a brightness change, rather than a periodic stream of frames. Due to the asynchronous and continuous nature of the data, established approaches cannot be applied to event-based cameras. In fact, visual-SLAM with event-based cameras has not been demonstrated and remains an open research problem. However, event-based cameras have a great potential in terms of energy usage (if nothing changes in the scene, the camera sends no data), dynamic range, and temporal resolution. It remains an open question which sensor should be selected, how they should be mounted on

a robot, and which algorithms should be applied such that the robot can achieve a certain task with the required reliance and energy usage, in a given environment. We face a design problem that is a tuple of "functionality space", "implementation space", and a "resource space", which should be addressed in an optimization [Censi et al., 2015, Censi, 2015].

Denser Visual-SLAM From an information-theoretic perspective, using every pixel in an image for motion estimation must be optimal. Indeed, previous work on dense tracking and mapping has for instance resulted in impressive performance in the presence of motion blur [Newcombe et al., 2011b] or dynamic contents [Newcombe et al., 2015]. However, optimal estimation of dense structure and motion with consistent estimation of the uncertainties in the system is still an open research problem. Therefore, novel map representations are required, which capture the correlations between the uncertainties in the dense surface with the uncertainty in the camera poses, similar to the sparse bundle adjustment problem.

Scene Understanding In terms of dense mapping, it is highly promising to tightly couple scene understanding with reconstruction [Salas-Moreno et al., 2013]. Knowing the object of interest allows us to use class-specific priors [Häne et al., 2013], while having a 3D reconstruction improves object detection and classification.

Understanding the semantics of the environment is also very relevant for MAV applications. For instance, knowing the surface properties helps in selecting suitable landing spots. On the other hand, detection, tracking, and activity classification of humans is paramount to enable truly interactive behavior of MAVs with humans. This is for example necessary for any "autonomous cinematography drone" that creates professional videos and photos of people in scenic places or performing sport activities. Such a drone must understand the scene to select the best view-points.

Scaling Visual SLAM Cars, cameras, and smart-phones of today are GPS enabled. In the future, these kind of devices may well be *visual SLAM enabled*, meaning that they always estimate their precise location, both indoors and outdoors by using tiny integrated cameras. To obtain local maps, these devices may be connected to a cloud back-end that stores and continuously updates a map of the world. Realizing this vision will require significant advancements in long-term visual mapping capabilities [Churchill and Newman, 2013], map compression [Dymczyk et al., 2015], and distributed data management [Cieslewski et al., 2015].

Robotics for Vision Finally, a very promising research direction is the tight integration of vision and robot control [Soatto, 2009]. Humans for instance do not run at full pace from bright sunlight into unknown dark terrain. Instead, we slow down, knowing that our eyes need time to adjust to different illumination conditions. Equally, in the rare case when our eyes are tricked by scene ambiguities, we change the vantage point to resolve the uncertainties. This is a powerful capability that has a great potential to increase the robustness of the overall robotic system. If a robot is aware that VO is unreliable in areas with little texture, it will try to select a trajectory that avoids such regions. Similarly, if an MAV has to perform an acrobatic maneuver, it should select its orientation such that motion blur in the recorded images is minimized. Any of such active behavior requires full knowledge of the uncertainties in the sensor measurements and it is necessary to have a consistent estimate of the uncertainties in the state of the system.

A Collaborative Monocular SLAM

C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza. Collaborative monocular SLAM with multiple micro aerial vehicles. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3962–3970, 2013. URL http://dx.doi.org/10.1109/IROS.2013.6696923.

Collaborative Monocular SLAM with Multiple Micro Aerial Vehicles

Christian Forster, Simon Lynen, Laurent Kneip, Davide Scaramuzza

Abstract — This paper presents a framework for collaborative localization and mapping with multiple Micro Aerial Vehicles (MAVs) in unknown environments. Each MAV estimates its motion individually using an onboard, monocular visual odometry algorithm. The system of MAVs acts as a distributed preprocessor that streams only features of selected *keyframes* and relative-pose estimates to a centralized ground station. The ground station creates an individual map for each MAV and merges them together whenever it detects overlaps. This allows the MAVs to express their position in a common, global coordinate frame. The key to real-time performance is the design of data-structures and processes that allow multiple threads to concurrently read and modify the same map. The presented framework is tested in both indoor and outdoor environments with up to three MAVs. To the best of our knowledge, this is the first work on real-time collaborative monocular SLAM, which has also been applied to MAVs.

Introduction

Motivation

Micro aerial vehicles will soon play a major role in missions, such as security surveillance, search and rescue, and environment inspection. However, for such operations, navigating based on GPS information only is not sufficient. Fully autonomous operation in urban environments and indoor spaces requires micro helicopters to rely on alternative localization systems. However, weight restriction and battery autonomy impose great limitations on the choice of the sensors. For small-sized and lightweight

platforms (less than 40cm and less than 1kg), laser scanners are still too heavy and consume too much power. Therefore, the only viable solution is to use a combination of onboard cameras and IMU (Inertial Measurement Unit). Successful demonstrations of a MAV performing autonomous basic maneuvers, using only a single onboard camera, IMU, and an onboard Atom computer, have been done in our previous work [Bloesch et al., 2010, Weiss et al., 2011b]. In this paper, we attempt to go one step forward, and address the problem of collaborative localization and mapping with multiple MAVs in unknown environments.

The application to multiple agents allows the use of redundant and parallel mechanisms to achieve increased robustness and efficiency. Several tasks—such as the workload of mapping an environment—can be shared among all the agents. As a practical result, the shared map among the robots allows the computation of the relative configuration of the agents, which forms a basis for multi-robot path planning and cooperative behaviors. Despite these advantages, solving the Simultaneous-Localization-And-Mapping (*SLAM*) problem with multiple robots generally increases the computational and inter-robot communication load.

Related Work

Most works in multi-robot SLAM have been done using range sensors (e.g., laser, sonars, and stereovision) and/or ground mobile robots moving in the same 2D plane [Fox et al., 2000, Rocha et al., 2005, Howard, 2006, Trawny et al., 2009]. Very little work has been done using bearing-only sensors (monocular vision) and for unconstrained (6DoF) motion of the agents (e.g., wearable sensors, hand-held cameras, and flying robots). This problem—known as multi-camera structure from motion or multi-camera SLAM—can be approached differently depending on whether the cameras (i.e., the robots) can "see" each other or not. If the former case, their relative configuration can be inferred from the relative bearing-angle observations [Martinelli et al., 2005, Cognetti et al., 2012]. In the latter case, this can be done starting from the common scene observed by the cameras. The work described in this paper belongs to the second category.

In [Sola et al., 2008], the authors use a single extended Kalman filter SLAM algorithm with an extended state vector composed of each camera pose and the observed features. Specifying the relative configuration at startup, they demonstrate results on two cameras attached to two bicycles. In [Vidal-Calleja et al., 2011], the authors describe a system for cooperative mapping using both aerial and ground robots equipped with stereo cameras. Each robot creates local submaps using an extended Kalman filter and maintains a global graph of submap positions. Rendezvous between robots, feature correspondences, and absolute GPS localization measurements, trigger loop closures which results in exchange of submap positions among the robots. In [Achtelik et al.,

2011], the authors study the case of two MAVs which, equipped with monocular cameras and IMU, form a flexible stereo rig. Using feature correspondence in the overlapping field of view, the relative pose of the two robots can be estimated. In [Danping and Ping, 2012], the authors process the video streams from multiple hand-held cameras. The process is synchronized in that the images from all the cameras are processed all at once at each time step. This makes their system impractical for robotic applications, where the input of each camera should be computed asynchronously in order to cope with missing data and delays. Additionally, it is assumed that all cameras observe the same scene at start. In [McDonald et al., 2011], a system was presented, where a single robot has to continuously localize within maps created during previous mapping sessions by the same robot. Although this work was not applied to multiple robots, it can, however, be seen as an instance of a multi-robot mapping process where each map was created in previous sessions by the same robot. Finally, in [Cunningham et al., 2013], a fully decentralized SLAM system is presented where each robot maintains a consistent augmented local map that combines local and neighbourhood information. The system has been validated in simulation.

Contributions and Outline

In the endeavor of enabling multi-robot navigation of MAVs with very-low onboard computing power, our goal is to employ the MAV onboard computer for low-level tasks—such as feature extraction, relative-motion estimation, and flight control—and delegate a ground station to higher-level tasks—such as mapping, loop-closure detection and map merging. The decoupling of motion estimation and mapping is useful in real-world scenarios, where the robots have to maintain some degree of autonomy in case of intermittent communication with the ground station.

An overview of the proposed approach is depicted in Figure A.1(a). Each MAV estimates its motion individually by running an onboard visual odometry (VO) algorithm that is used to both track the robot motion and stabilize its 6DoF pose during flight. The outputs of the VO—i.e. keyframe features and relative-pose estimates between keyframes—are streamed to a central ground station where our *Collaborative Structure from Motion* (CSfM) system is running. The CSfM system on the ground station creates an individual map for each MAV and merges them together whenever it detects overlaps. The ground station processes the data asynchronously, as it arrives, which accounts for situations where the robots do not start all at the same time or where some data are missing due to a communication failure. To achieve real-time performance, we design data-structures and processes that allow multiple threads (one for each MAV) to concurrently read and modify the same map. Additionally, we devise a solution to tackle the scale-difference between the onboard-estimated trajectories and those estimated on the ground station.

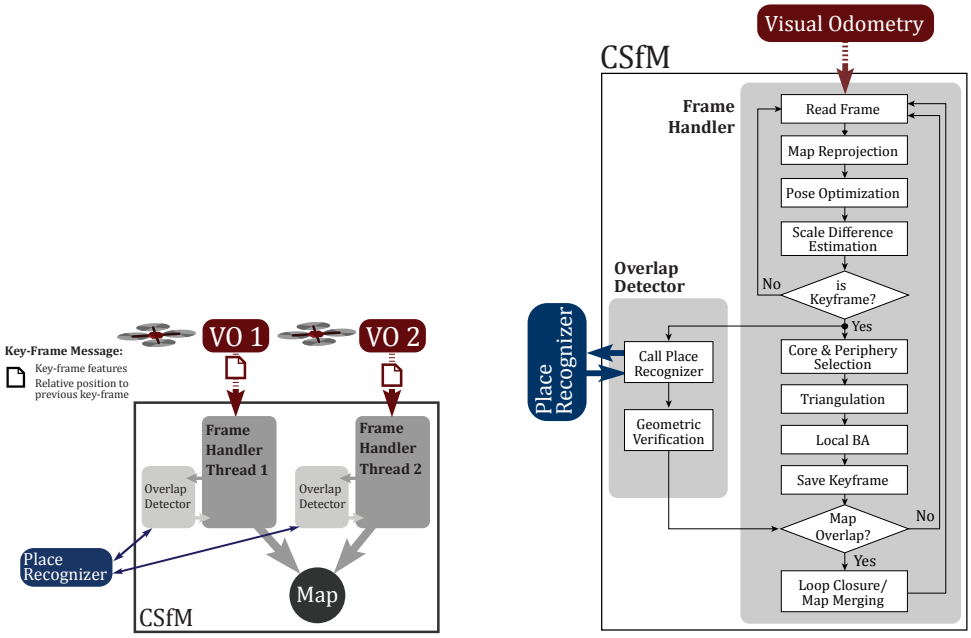


Figure A.1 – (a) The CSfM system—running on the ground station—creates a separate thread for each MAV. Initially, each thread creates its own map. However, the maps are merged when the Place Recognizer detects an overlap between the two maps. Both threads then read and update the same map simultaneously. (b) Mapping pipeline executed inside the frame-handling threads of the CSfM system.

The remainder of the paper is structured as follows. Section A.2 provides an overview of the CSfM system. Section A.3 details the general mapping pipeline. Section A.4 explains how overlaps between maps are detected and how they are merged into a single global map. Section A.5 describes the implementation design for concurrent map access. Finally, Section A.6 provides the experimental results.

System Overview

Each MAV tracks its own position using a keyframe-based onboard monocular VO algorithm. We chose to employ the VO presented in our previous work [Kneip et al., 2011a]. It is boosted in terms of robustness and efficiency through the use of the relative-rotation prior from the onboard IMU. However, our proposed CSfM system is modular and, therefore, any alternative keyframe-based VO algorithm (such as [Klein and Murray, 2007]) could be used.

Figure A.1(a) illustrates how the CSfM system is embedded in the multi-robot mapping framework. For each MAV, the CSfM system (running on the ground station) creates a new *frame-handler* thread that receives directly the keyframe messages from the corresponding VO. The frame handler creates a new map for its MAV and processes

the received keyframe messages in parallel and asynchronously to the other frame-handler threads. A keyframe message only contains the extracted image features (i.e., image coordinates and descriptors) along with a relative transformation to the previous keyframe.

A keyframe is only added to a map when it provides new information. Right after a frame handler decides to add a keyframe to the map, it passes it on to its own overlap-detection thread (see Figure A.1(b)). The overlap detectors in turn pass the keyframes on to a *place-recognition module*. The place recognizer accumulates the visual information (i.e., feature descriptors), from all keyframes in every map, and quickly detects whether a place has been visited before. Meanwhile, the frame handlers triangulate new points and perform local *Bundle Adjustment* (BA) in the current keyframe's neighborhood. If the overlap detector detects an overlap within the map of the same MAV, a loop-closure optimization is initiated. Conversely, if the overlap occurs with the map of another MAV, the affected frame handlers are temporarily suspended to allow merging of the maps into a single one.

After map merging, the frame handlers operate on the merged map. Specially-designed data structures and the use of C++ concurrency-control mechanisms allow multiple frame-handler threads to safely access and update the common map, which is also the key to real-time performance.

Mapping Pipeline

Figure A.1(b) illustrates the mapping pipeline as implemented in the frame handler. The following sections detail the individual building blocks.

Keyframe Message

Each MAV tracks its own position (with respect to its own starting point) using a keyframe-based onboard monocular VO algorithm. When the onboard VO selects a new keyframe, a message to the ground station is sent containing the extracted features along with the relative transformation $(\hat{\mathrm{R}}_{k-1,k}, \hat{\mathbf{p}}_{k-1,k})$ to the previous keyframe.

Handling the Keyframe Message by the Ground Station

When the ground station receives a keyframe message from a MAV, there are two possibilities: (i) if this is the first message from that MAV, then the CSfM system (running on the ground station) creates a new frame-handler thread and triangulates the received features into map-points as soon as the next message arrives; (ii) if a frame-handler for that MAV already exists, correspondences between the existing 3D

map-points and the features in the new keyframe are identified. Additionally, the frame-handler updates the absolute pose $(\mathbf{p}_k, \mathrm{R}_k)$ of the new keyframe in the map:

$$\mathbf{p}_k = \mathbf{p}_{k-1} + \mathrm{R}_{k-1}\, \hat{\mathbf{p}}_{k-1,k}, \tag{A.1}$$

$$\mathrm{R}_k = \mathrm{R}_{k-1}\, \hat{\mathrm{R}}_{k-1,k}\ . \tag{A.2}$$

Pose Optimization

The CSfM system optimizes the absolute 6DoF pose of the new keyframe within the map by minimizing the reprojection error of all map-points visible by that keyframe using a nonlinear least-squares solver [Kümmerle et al., 2011].[1]

Scale-Difference Estimation between VO and CSfM

Each MAV's onboard monocular VO produces motion and structure information only up to an unknown scale factor. Furthermore, this scale factor is not constant, but drifts over time. On the ground-station side, the CSfM system also exhibits a scale drift as long as no loop closures occur. These two scale factors are not equal and diverge at *different rates* (see Figure A.2). If the scale difference is not corrected, scale jumps can occur as it is depicted in Figure A.5 (refer to Section A.6). A scale jump occurs if the MAV's VO's scale drifts too much with respect to the CSfM map such that in the reprojection step no correspondences can be found and thus the pose cannot be optimized anymore towards the right position.

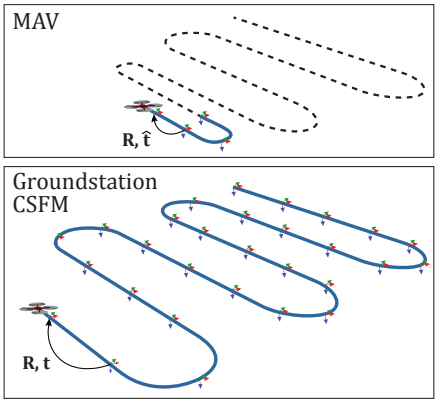


Figure A.2 – The VO on the MAV maintains a map with a limited number of keyframes (e.g., 5) for processing-speed reasons. Therefore, the scale of the onboard VO drifts faster than on the ground-station. The relative translation $\hat{t}$ computed by the MAV's onboard VO needs to be corrected with the scale factor λ for the CSfM map.

To correct this scale difference, we compare the estimated relative translation $\mathbf{p}$ before and after the pose optimization step:

$$\hat{\lambda}_k = \frac{||\mathbf{p}_{k-1,k \text{ after Optimization}}||}{||\hat{\mathbf{p}}_{k-1,k \text{ before Optimization}}||}. \tag{A.3}$$

[1] The *reprojection error* is the Euclidean distance e between the reprojected point and the corresponding observed feature in the image plane.

Hence, we compute the new scale factor λ_k with the following update rule:

$$\lambda_k = \lambda_{k-1} + \kappa \cdot (\hat{\lambda}_k - \lambda_{k-1}), \tag{A.4}$$

where κ represents the smoothing factor. Empirically, we found that $\kappa = 0.05$ is a good choice.

Using the estimated scale-difference factor λ_k, the relative position received from the MAV's onboard VO is corrected by the corresponding frame-handler before a new keyframe is used. The position computed by Equation (A.1) is then updated to:

$$\mathbf{p}_k = \mathbf{p}_{k-1} + \lambda_k \mathrm{R}_{k-1}\, \hat{\mathbf{p}}_{k-1,k}. \tag{A.5}$$

This step further justifies why the pose of keyframes that are later not inserted in the map must also be optimized. It allows us to successfully track the robot's pose with respect to the map and to estimate the scale difference.

Keyframe Selection

While the use of more map-points improves the accuracy of the map, increasing the number of keyframes has only minor effects once robustness is achieved [Strasdat et al., 2010]. Therefore, similar to [Klein and Murray, 2007], new keyframes are only inserted in the map if the distance to the closest keyframe is large enough.[2] Depending on the trajectory, the CSfM system rejects on average up to 85% of the received keyframes, which saves processing time.

Selection of Core and Periphery Keyframes

The CSfM system follows the fundamental concept that no temporal ordering of keyframes is retained. Keyframe neighbourhoods for optimization and triangulation are selected based only on spatial adjacency. This means that also older keyframes—regardless of the MAV they originate from—are taken into account for these operations, leading to a reduction of redundant information inside the map. A set $\mathcal{C}$ of four *core keyframes* is selected, which shares the largest number of common map-point observations with the new keyframe. The set of *periphery keyframes* $\mathcal{P}$ is defined by all keyframes that share at least one common map-point observation with $\mathcal{C}$ or with the new keyframe but which are not in the set $\mathcal{C}$.

[2] We set the threshold to 15% of the average scene depth.

Triangulation

New map-points are triangulated when a new keyframe is selected to be inserted in the map. For every unmatched feature in the new keyframe, we search matching features along the epipolar lines in the core keyframes. If a matching descriptor is found, the point is triangulated and projected into the remaining *core* and *periphery* keyframes to increase the number of measurements. The creation of duplicate points is inhibited by merging points in case a feature is already associated with an existing map-point. The merging step is essential for obtaining sparse and well constrained maps.

Local Bundle Adjustment

Mouragnon et al. [2006] have shown the feasibility of creating an accurate 3D reconstruction in real-time using incremental *bundle adjustment*. Therefore, the CSfM system optimizes the set of *core* keyframes $\mathcal{C}$ together with the new keyframe and along with the commonly observed map-points using the g^2o framwork [Kümmerle et al., 2011]. The set of periphery keyframes $\mathcal{P}$ is added to the optimization window with a fixed pose. The periphery keyframes are required to fix the scale of the structure and to ensure that the optimization is optimal with respect to the boundary.

Map Overlap Detection and Processing

A fundamental characteristic of the CSfM system is its ability to detect if a MAV reenters an environment that has already been visited, either by itself or by another MAV which results in a loop-closure optimization or a map merging respectively. Such overlaps are detected based on the keyframe appearance (i.e., feature descriptors) and subsequently geometrically verified.

Appearance-based Overlap Detection

If a keyframe is accepted for inclusion in the map, a second overlap-detection thread is started, which calls the *place-recognizer* module (see Figure A.1(b)). The external place recognition module is the same for all frame handlers and relies on a *bag-of-words* [Sivic and Zisserman, 2003] approach. The exact type of place recognizer in use depends on the employed local invariant point descriptor. We initally tested OpenSURF features [Evans, 2009], which allow the use of the OpenFABMAP place recognizer [Glover et al., 2010]. However, for increased speed, we decided to use BRISK features [Leutenegger et al., 2011]. Since binary features have special clustering properties, a dedicated place-recognition module was implemented.[3]

[3]The BRISK-based place-recognizer goes beyond the scope of this paper and, therefore, it is not described here.

Geometric Verification

Each time the place recognizer returns an *overlap-keyframe* with similar appearance as the current keyframe, the overlap detector geometrically verifies the result by applying the Perspective-Three-Point (P3P) algorithm from our previous work [Kneip et al., 2011b]. The P3P algorithm derives the camera pose from at least three 3D-to-2D feature correspondences. These correspondences are established by identifying matching descriptors between map-points—which the *overlap-keyframe* observes—and features in the current keyframe. To remove outliers, we integrated the P3P into a RANSAC [Fischler and Bolles, 1981] procedure. The output of RANSAC is then the rigid body transformation between the two keyframes.

Map merging

If the detected overlap occurred between two different maps, the similarity transformation $\{\mathbf{R}, \mathbf{t}, s\}$ returned by the geometric verification step is used to merge the two maps into one. The factor s accounts for the different scale between the two maps and can be found by comparing the relative distances between any combination of 3D map-points which are common between the two maps. All frame handlers working on either of the two maps are temporarily suspended, and the entire candidate map for which an overlap was detected is subjected to the determined similarity transformation. To improve the measurements of points and avoid redundant information in the map, all map-points from each overlapping map region are reprojected into the keyframes from the other map and corresponding map-points are merged. A last important detail consists of applying the scale factor s to the scale difference factor (see Section A.3.4) of all frame handlers that were operating on the transformed map. This is necessary to ensure that the received relative position estimates from the VO are correctly scaled with respect to the map. The frame-handler threads are finally resumed, and now operate in parallel on the same map. At this stage, it is important to design the algorithm and data structures such that concurrent data access is possible (see Section A.5).

Note that the CSfM node creates references between two maps only when a loop closure is detected. However, in practice, the two maps may still contain overlaps in other regions if the place-recognition or the geometric-verification steps failed to detect them earlier. However, the CSfM system is still able to detect and incorporate them in a later stage in case a MAV retraverses the same environment.

Loop closure

The computed similarity transformation parameters $\{\mathbf{R}, \mathbf{t}, s\}$ also incorporate the amount of drift that has been accumulated along the loop.

The standard solution to optimize both the full map and keyframes after loop closure is to run global BA. However, this approach is computationally demanding and may fail completely due to convergence into local minima. Therefore, we chose to split the optimization into two steps. In the first step, we marginalize out the map-points. This reduces the map representation to a pose-graph with edges of different *strength* between poses. Strasdat et al. [Strasdat et al., 2010] were the first to propose 7-DoF pose-graph optimization including the scale as a drift parameter, which leads to a substantial improvement in a monocular-SLAM context. The parametrization of this pose-graph relaxation is included in the g^{2}o framework [Kümmerle et al., 2011] and used by the CSfM system. After pose-graph optimization, the map-points are updated accordingly and global BA is run to further refine both map-points and keyframe poses simultaneously.

Implementation Design for Concurrent Map Access

If two or more maps have been merged, multiple frame-handling threads concurrently read and modify a single map (as depicted in Figure A.1(a)). Processing keyframes in parallel on a multi-core processor is the key to real-time performance of the CSfM system. However, when multiple things happen at the same time, special measures need to be taken both on the data-structure and on the algorithm layout level.

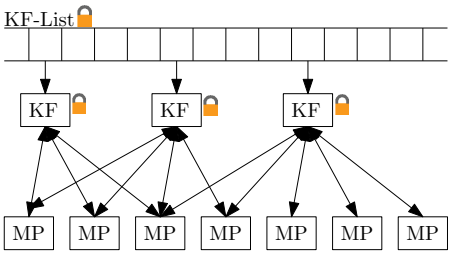


Figure A.3 – Data-structure design for concurrency.

The difficulty of *shared memory* between multiple threads comes from the consequences of modifying data. We can ensure the integrity of the shared data by using the concept of *mutual exclusion locks*. This concept defines that if a thread wants to access some data-object, it first needs to acquire the data-object's lock, which is only possible if no other thread has previously acquired the lock without releasing it.

The design of data-structures defines the scope for simultaneous data access. Figure A.3 illustrates the map data structure of the CSfM system. The map consists of a list of keyframes (`KF-List`), whereas each keyframe (`KF`) holds a list of references to map-points (`MP`) that it observes. The map-points in turn also have a list of references to keyframes which they are observed by. There are only locks on the keyframe list and on the individual keyframes. If a thread owns the lock of a keyframe, it is allowed to read all map-points which the keyframe observes. Hence—in order to modify a map-point—it is necessary to acquire the locks of all keyframes that observe the map-point.

This property is used in the mapping pipeline: If a frame-handler thread locks all core and periphery keyframes $\{\mathcal{C}, \mathcal{P}\}$, it is allowed to modify these keyframes and all map-points which are observed by at least one of the core-keyframes $\mathcal{C}$. Fortunately, this is exactly the set of objects which change during local BA and triangulation.

Since no list of map-points exists, all map-points must be accessed via a keyframe. Knowledge of the lock state of this keyframe automatically inhibits that threads modify data that are out of their scope. Moreover, this design eliminates the *overhead* of locking individual map-points.

The employed locking strategy uses *shared* and *upgradeable locks*[4] which allows other threads to simultaneously read the data in the same neighborhood except for the negligible time when updates are saved in the map. If the MAVs operate in different parts of the map with non-overlapping core and periphery keyframes, they can even update the map concurrently.

Experimental Results

Experiments were performed using two AscTec FireFly MAVs[5] equipped with an IMU, a single downlooking camera, and a Core-2-Duo computer. The ground-station was a 2.8 GHz i7 laptop. A video of the experimental results is available at http://rpg.ifi.uzh.ch.

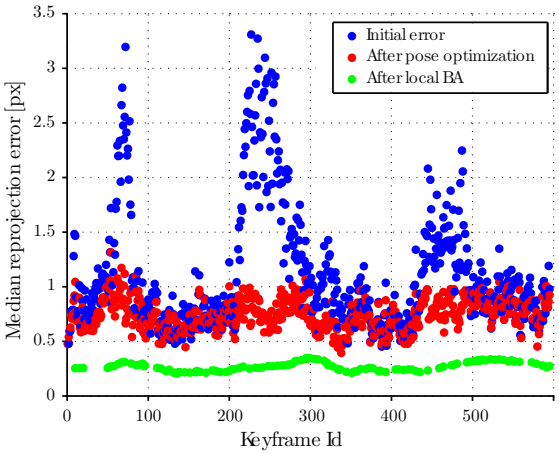


Figure A.4 – Influence of pose optimization and local BA on the reprojection error in newly received keyframes.

To evaluate its performance, our CSfM back-end was tested in both indoor and outdoor environments. Indoors, ground truth was obtained from a Vicon motion-capture system that provides absolute position information with millimeter accuracy at 100 Hz. The output of the CSfM was evaluated by comparing the keyframe positions to the ground truth.[6] The indoor environment consisted of a flat surface of approximately 8 by 8 meters. We added additional texture to the surface such that the VO algorithm has always enough features to track.

[4] Implemented e.g. in the Boost library: `www.boost.org`.

[5] `www.asctec.de/Firefly`

[6] Since the scale of the map created by the CSfM system and the coordinate transformation between the computed and the ground truth trajectories are unknown, we derived the aligning similarity transformation $\{\bar{\mathbf{R}}, \bar{\mathbf{t}}, \bar{s}\}$ using a least-squares procedure.

(a) Without scale correction. Scale jump is marked with a blue arrow.

(b) With scale correction.

Figure A.5 – Two maps of a flat surface seen from the side. The points on the bottom represent the map-points and the triangles on the top the key-frames. In (a), the scale jump of the map is clearly visible (blue arrow). In (b), the scale-difference estimation was activated and no scale-jump occurs.

Figure A.4 illustrates the influence of pose optimization (Section A.3.3) and local BA (Section A.3.8) on the median reprojection error of map-points. The peaks in the initial reprojection errors (blue dots) originate from the scale difference discussed in Section A.3.4. As observed, they get canceled after pose optimization and local BA. Moreover, the peaks in the initial error disappear as soon as the scale difference factor λ_k has adapted. Figure A.5 shows the effect of scale difference estimation and correction that we mentioned in Section A.3.4. If the scale drifts, the system is able to recover for reprojection errors up to 5 pixels. However, when the drift becomes too large, the system looses connection to the map and scale jumps occur. This effect does not arise if the scale is estimated and corrected, as shown in Figure A.5b. The amount of drift depends on the chosen trajectory and on the distribution of the features in the keyframes.

Figure A.6(a) shows a large loop trajectory before and after pose-graph optimization. On this trajectory, the system adopted 110 keyframes over 16.7m. The evolution of the error over time is reported in Figure A.6(b). The loop closure occurs around 35s and measurements are only indicated at times when a keyframe was created. The RMS error of the keyframe positions right before and after loop-closure detection and optimization was 0.1m and 0.04m, respectively.

Figure A.7a shows two MAVs simultaneously mapping two distinct areas. As soon as the CSfM system detects an overlap (b), it merges the two maps into a single, global one. Figure A.7(c) shows comparison with ground truth obtained from a Vicon motion-capture system. Figure A.7(d) indicates the corresponding error for both trajectories. The length of the combined trajectory was 30m, the total number of keyframes in the final map was 154, and the final RMS error 0.06m.

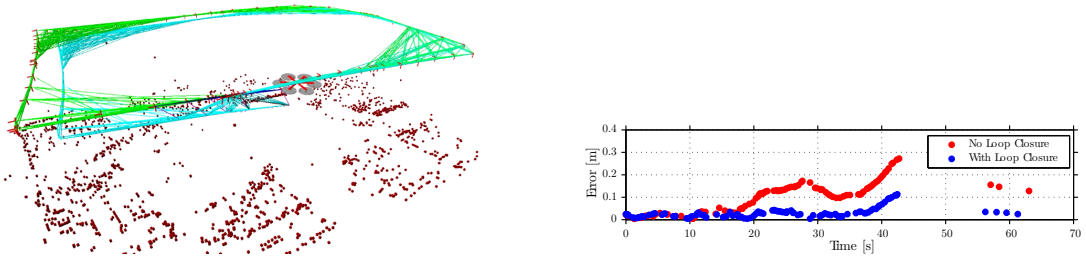
Our CSfM algorithm was also tested on two outdoor datasets from the European project *sFly* [Scaramuzza et al., 2014]. The combined trajectory length was approximately 400m (see Figure A.8(e)). Figures A.8(a) to A.8(d) show the mapping of the outdoor environment. Since both MAVs start at the same location, the two maps are immediately

merged. Hence, the relative pose of the two MAVs is known from the beginning. Based on the color of the map-points, which is set to the color of the last MAV that observed it, one can see that both MAVs successfully localize in parts mapped by the other MAV (e.g., compare Figures A.8(c) and A.8(d)). The GPS accuracy around the test area was ranging between 5 and 15 meters because of foliage and surrounding buildings. Therefore, GPS cannot be used as a reliable ground-truth. Nevertheless, some drift is still clearly visible between the estimated trajectory and the GPS. This is due to the absence of loop closures between the trajectories undergone by the two MAVs.

The most common failure case of the system occured when the place-recognition module missed to detect an overlap. In this case, after merging at a later stage, the global map contained redundant and, because of drift and map alignment errors, slightly misaligned map-points. The system was often able to recover from such situations through loop-closure detection and optimization at a second traversal.

By transmitting only binary features extracted from keyframes, the required bandwidth can be kept at a considerably low level ($\sim$1 Mbit/s for 200 BRISK [Leutenegger et al., 2011] features and 10 Hz keyframe rate) compared to streaming entire raw images ($\sim$86.6 Mbit/s for grayscale 752$\times$480-pixel images and 30 Hz framerate). Note that the reduced keyframe-rate for our approach is because our VO already preselects a subset of frames as keyframes.

The average keyframe processing time on the ground-station ranged between 22ms and 45ms, resulting in a frame rate of up to 45Hz for one MAV. The average computation time per keyframe depends, to a large extend, on both the trajectory and the environment. A MAV that is constantly exploring new environments produces more keyframes—and, thus, a higher workload for the CSfM system—than a MAV that remains in previously-mapped areas. In the latter situation, most keyframes are dropped since they do not provide new information. On average the CSfM system selects only 85% of all received keyframes. Regarding the environment, an increased density of features implicitly leads to an increased number of map-point references and, thus, higher reprojection, matching and BA computation times. The average number or reprojected map-points ranged between 200 and 350. Furthermore, the efficiency of the algorithm also depends on inherent parameters, such as keyframe-selection criteria and size of the local BA window. The parallelized system pipeline is designed such that the processing time does not increase with higher numbers of MAVs—given that for each MAV a processing core is provided. In experiments on the mentioned 4-core laptop, the processing time with two MAVs did not decrease significantly and with three MAVs real-time performance could still be achieved. One bottleneck is the place recognition module which currently is not parallelized and sequentializes the requests.



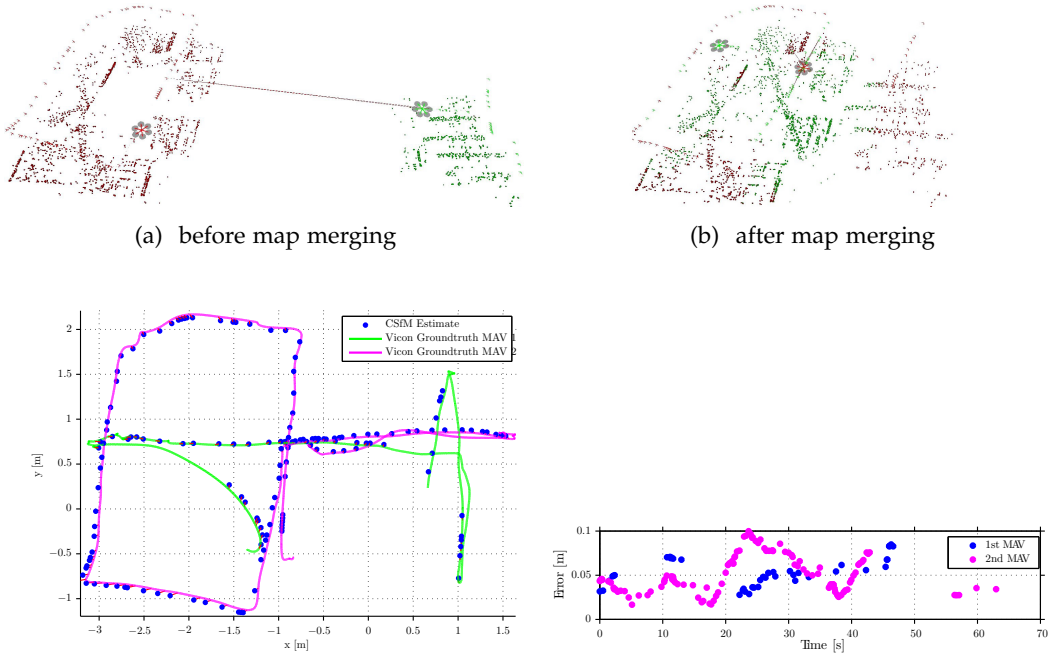
(a) (b)

Figure A.6 – (a) 3D visualization of a pose-graph before (blue) and after (green) optimization on a loop trajectory. (b) Evolution of the RMS error of the loop trajectory (a) with and without explicit loop closure optimization. The loop closure occurs around 35 s.

Conclusion and Future Work

We proposed a system (named CSfM) for collaborative monocular SLAM with multiple MAVs using a centralized approach. By distributing the workload between the MAVs and the ground-station, we save processing power, require much less transmission bandwidth, and keep some autonomy on the MAVs themselves, i.e. the stability of the MAVs is not threatened by the reliability of the communication link. The CSfM system is highly modular and can work with different VO and place recognizer modules. We also presented a method for scale-difference correction, which solves an inherent problem of the decoupled system. The algorithm employs state-of-the-art techniques for active loop closure detection, bundle adjustment, and 7-DoF pose-graph relaxation. Results on real data including a comparison to ground truth demonstrate the high accuracy that can be achieved with vision-only SLAM. Finally, real-time performance was achieved with a system that allows multiple threads to concurrently read and modify the same map.

Future work will leverage on the potential to localize multiple MAVs in the same environment to allow purely vision-based coordinated flight of multiple robots.



(a) before map merging

(b) after map merging

(c) before map merging

(d) RMS error

Figure A.7 – Experimental results showing maps concurrently created by two MAVs in a real indoor environment. (a) The maps shortly after an overlap was detected by the place recognizer (red line). (b) The global map after merging. (c) The map of Figure A.7b (after loop-closure and map-merging) is compared to the ground-truth. The blue dots mark the keyframe positions, while the green and purple lines are the ground-truth trajectories of both MAVs. (d) Evolution of the RMS error of the keyframes.

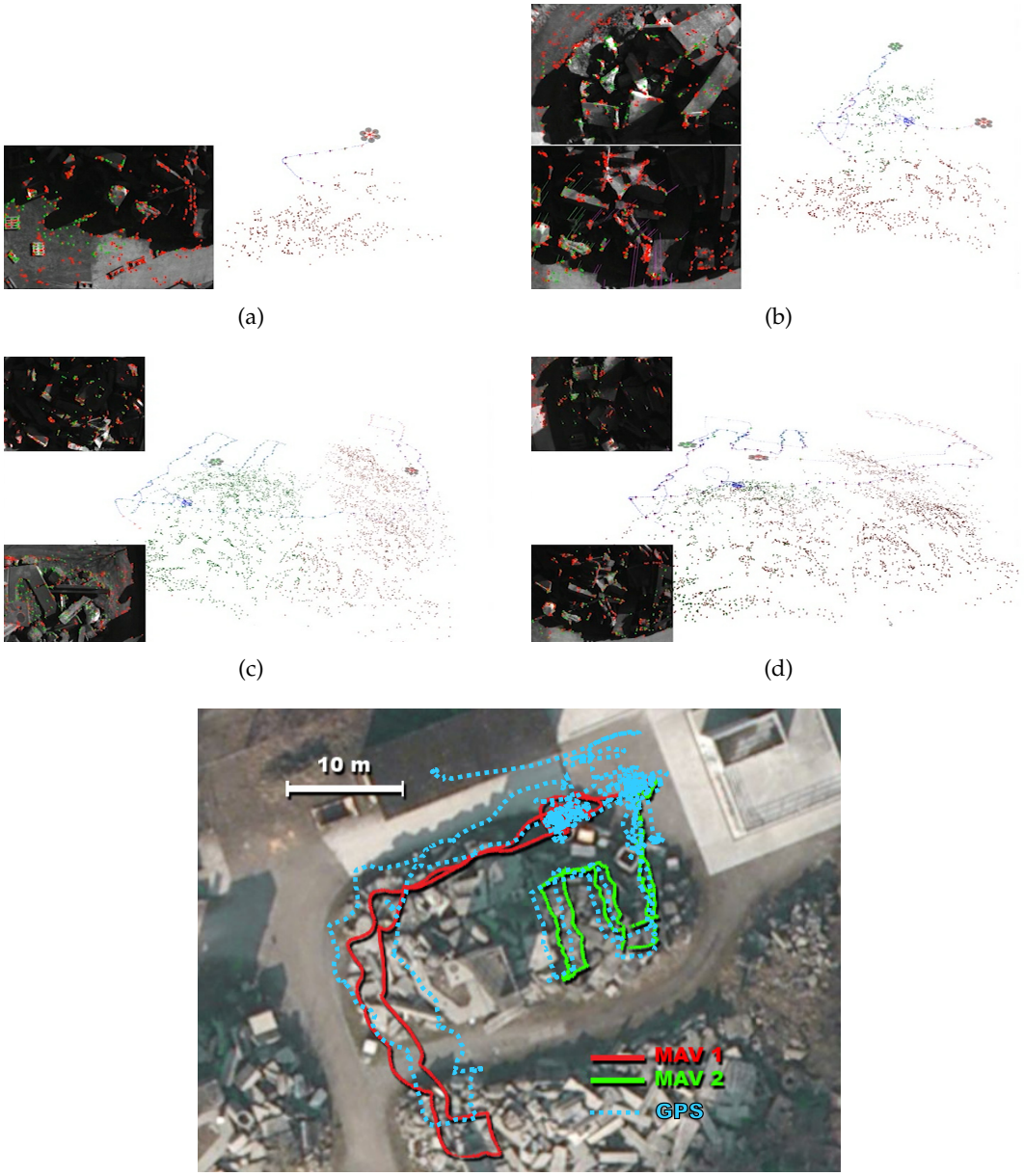


Figure A.8 – (a) The first MAV takes off and starts building its map. (b) The second MAV starts and immediately localizes in the map of the first MAV. The relative position of the two MAVs is now known. (c) The MAVs return to the take-off location. (d) Note that the color of the map-points indicate which MAV has last observed the points. One can observe that the red MAV observes and localizes with the map-points created by the green MAV. (e) Trajectories of the two MAVs in the outdoor experiment overlayed with the GPS measurements.

B Semi-Direct Visual Odometry

This chapter is a reprint of the article currently under review as:

C. Forster, Z. Zhang, M. Gassner, M. Werlberger, D. Scaramuzza. Semi-Direct Visual Odometry for Monocular and Multi-Camera Systems. *IEEE Transactions on Robotics (TRO), 2016*.

A shorter version of this article was previously published as:

C. Forster, M. Pizzoli, and D. Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 15–22, 2014. URL http://dx.doi.org/10.1109/ICRA.2014.6906584.

Semi-Direct Visual Odometry for Monocular and Multi-Camera Systems

C. Forster, Z. Zhang, M. Gassner, M. Werlberger, D. Scaramuzza[1]

Abstract — Direct methods for Visual Odometry (VO) have gained popularity due to their capability to exploit information from all image gradients in the image. However, low computational speed as well as missing guarantees for optimality and consistency are limiting factors of direct methods, where established feature-based methods instead succeed at. Based on these considerations, we propose a Semi-direct VO (SVO) that uses direct methods to track and triangulate pixels that are characterized by high image gradients but relies on proven feature-based methods for joint optimization of structure and motion. Together with a robust probabilistic depth estimation algorithm, this enables us to efficiently track pixels lying on weak corners and edges in environments with little or high-frequency texture. We further demonstrate that the algorithm can easily be extended to multiple cameras, to track edges, to include motion priors, and to enable the use of very large field of view cameras, such as fisheye and catadioptric ones. Experimental evaluation on benchmark datasets shows that the algorithm is significantly faster than the state of the art while achieving highly competitive accuracy.

Introduction

Estimating the six degrees-of-freedom motion of a camera merely from its stream of images has been an active field of research for several decades [Ullman, 1979, Tomasi and Kanade, 1992, Chiuso et al., 2002, Nister et al., 2004, Davison et al., 2007, Scaramuzza and Fraundorfer, 2011]. Today, state-of-the-art visual SLAM (V-SLAM) and visual odometry (VO) algorithms run in real-time on smart-phone processors and approach the accuracy, robustness, and efficiency that is required to enable various

[1]The authors are with with the Robotics and Perception Group, University of Zurich, Switzerland. Contact information: {forster, zzhang, gassner, werlberger, sdavide}@ifi.uzh.ch. This research was partially funded by the Swiss National Foundation (project number 200021-143607, “Swarm of Flying Cameras”), the National Center of Competence in Research Robotics (NCCR), the UZH Forschungskredit, and the SNSF-ERC Starting Grant.

interesting applications. Examples comprise the robotics and automotive industry, where the ego-motion of a vehicle must be known for autonomous operation. Other applications are virtual and augmented reality, which requires precise and low latency pose estimation of mobile devices.

The central requirement for the successful adoption of vision-based methods for such challenging applications is to obtain highest *accuracy* and *robustness* with a limited computational budget. The most accurate camera motion estimate is obtained through joint optimization of structure (*i.e.*, landmarks) and motion (*i.e.*, camera poses). For *feature-based methods*, this is an established problem that is commonly known as *bundle adjustment* [Triggs et al., 2000] and many solvers exist, which address the underlying non-linear least-squares problem efficiently [Dellaert and Kaess, 2006, Kaess et al., 2012, Agarwal et al., Kümmerle et al., 2011]. Three aspects are key to obtain highest accuracy when using sparse feature correspondence and bundle adjustment: (1) long feature tracks with minimal feature drift, (2) a large number of uniformly distributed features in the image plane, and (3) reliable association of new features to old landmarks (*i.e.*, loop-closures).

The probability that many pixels are tracked reliably, *e.g.*, in scenes with little or high frequency texture (such as sand [Maimone et al., 2007] or asphalt [Lovegrove et al., 2011]), is increased when the algorithm is not restricted to use local point features (*e.g.*, corners or blobs) but may track edges [Klein and Murray, 2008] or more generally, all pixels with gradients in the image, such as in dense [Newcombe et al., 2011b] or semi-dense approaches [Engel et al., 2014]. Dense or semi-dense algorithms that operate directly on pixel-level intensities are also denoted as *direct methods* [Irani and Anandan, 1999]. Direct methods minimize the *photometric error* between corresponding pixels in contrast to feature-based methods, which minimize the *reprojection error*. The great advantage of this approach is that there is no prior step of data association: this is implicitly given through the geometry of the problem. However, joint optimization of dense structure and motion in real-time is still an open research problem, as is the optimal and *consistent* [Bar-Shalom et al., 2001, Huang et al., 2010] fusion of direct methods with complementary measurements (*e.g.*, inertial). In terms of efficiency, previous direct methods are computationally expensive as they require a semi-dense [Engel et al., 2014] or dense [Newcombe et al., 2011b] reconstruction of the environment, while the dominant cost of feature-based methods is the extraction of features and descriptors, which incurs a high constant cost per frame.

In this work, we propose a VO algorithm that combines the advantages of direct and feature-based methods. We introduce the *sparse image alignment* algorithm (Sec. B.5), an efficient direct approach to estimate frame-to-frame motion by minimizing the photometric error of features lying on intensity corners and edges. The 3D points corresponding to features are obtained by means of robust recursive Bayesian depth estimation (Sec. B.6). Once feature correspondence is established, we use bundle

adjustment for refinement of the structure and the camera poses to achieve highest accuracy (Sec. B.5.2). Consequently, we name the system *semi-direct* visual odometry (SVO).

Our implementation of the proposed approach is exceptionally fast, requiring only 2.5 milliseconds to estimate the pose of a frame on a standard laptop computer, while achieving comparable accuracy with respect to the state of the art on benchmark datasets. The improved efficiency is due to three reasons: firstly, SVO extracts features only for selected keyframes in a parallel thread, hence, decoupled from hard real-time constraints. Secondly, the proposed direct tracking algorithm removes the necessity for robust data association. Finally, contrarily to previous direct methods, SVO requires only a sparse reconstruction of the environment.

This paper extends our previous work [Forster et al., 2014b], which was also released as open source software.[2] The novelty of the present work is the generalization to wide FoV lenses (Sec. B.7), multi-camera systems (Sec. B.8), the inclusion of motion priors (Sec. B.9) and the use of edgelet features. Additionally, we present several new experimental results in Sec. B.11 with comparisons against previous works.

Related Work

Methods that simultaneously recover camera pose and scene structure, can be divided into two classes:

Feature-based The standard approach to solve this problem is to extract a sparse set of salient image features (e.g. corners, blobs) in each image; match them in successive frames using invariant feature descriptors; robustly recover both camera motion and structure using epipolar geometry; and finally, refine the pose and structure through reprojection error minimization. The majority of VO and V-SLAM algorithms [Scaramuzza and Fraundorfer, 2011] follow a variant of this procedure. A reason for the success of these methods is the availability of robust feature detectors and descriptors that allow matching images under large illumination and view-point changes. Feature descriptors can also be used to establish feature correspondences with old landmarks when closing loops, which increases both the accuracy of the trajectory after bundle adjustment Triggs et al. [2000], Klein and Murray [2007] and the robustness of the overall system due to re-localization capabilities. This is also where we draw the line between VO and V-SLAM: While VO is only about incremental estimation of the camera pose, V-SLAM algorithms, such as [Mur-Artal et al., 2015a], detect loop-closures and subsequently refine large parts of the map.

[2] http://github.com/uzh-rpg/rpg_svo

The disadvantage of feature-based approaches is their low speed due to feature extraction and matching at every frame, the necessity for robust estimation techniques that deal with erroneous correspondences (*e.g.,* RANSAC [Fischler and Bolles, 1981], M-estimators [MacTavish and Barfoot, 2015]), and the fact that most feature detectors are optimized for speed rather than precision. Furthermore, relying only on well localized salient features (*e.g.,* corners), only a small subset of the information in the image is exploited.

In SVO, features are extracted only for selected keyframes, which reduces the computation time significantly. Once extracted, a *direct* method is used to track features from frame to frame, resulting in outlier-free and sub-pixel precise matches. Apart from well localized corner features, this allows tracking and mapping any pixel with non-zero intensity gradient.

Direct methods Direct methods estimate structure and motion directly by minimizing an error measure that is based on the image's pixel-level intensities [Irani and Anandan, 1999]. The local intensity gradient magnitude and direction is used in the optimization compared to feature-based methods that consider only the distance to a feature-location. Pixel correspondence is given directly by the geometry of the problem, eliminating the need for robust data association techniques. However, this makes the approach dependent on a good initialization that must lie in the basin of attraction of the cost function.

Using a direct approach, the six degree of freedom (DoF) motion of a camera can be recovered by *image-to-model alignment*, which is the process of aligning the observed image to a view synthesized from the estimated 3D map. Early direct VO methods tracked and mapped few—sometimes manually selected—planar patches [Jin et al., 2003, Benhimane and Malis, 2006, Silveira et al., 2008, Mei et al., 2008, Pretto et al., 2011]. By estimating the surface normals of the patches [Molton et al., 2004], they could be tracked over a wide range of viewpoints. In [Comport et al., 2010], the local planarity assumption was relaxed and direct tracking with respect to arbitrary 3D structures computed from stereo cameras was proposed. For RGB-D cameras, where a dense depth-map for each image is given by the sensor, dense image-to-model alignment was subsequently introduced in [Meilland et al., 2011, Tykkala et al., 2011, Kerl et al., 2013]. In conjunction with dense depth registration this has become the standard in camera tracking for RGB-D cameras [Meilland and Comport, 2013, Whelan et al., 2014, Handa et al., 2014, Whelan et al., 2015]. With DTAM [Newcombe et al., 2011b], a direct method was introduced that computes a dense depthmap from a single moving camera in real-time. The camera pose is found through direct whole image alignment using the depthmap. However, inferring a dense depthmaps from monocular images is computationally intensive and is typically addressed using GPU parallelism, such as in the open-source REMODE algorithm [Pizzoli et al., 2014]. Early on it was realized

that only pixels with an intensity gradient provide information for motion estimation [Dellaert and Collins, 1999]. In this spirit, a *semi-dense* approach was proposed in [Engel et al., 2013] where the depth is only estimated for pixels with high intensity gradients. In our experimental evaluation in Sec. B.11.1 we show that it is possible to reduce the number of tracked pixels even more for frame-to-frame motion estimation without any noticeable loss in robustness or accuracy. Therefore, we propose the *sparse* image-to-model alignment algorithm that uses only sparse pixels at corners and along image intensity gradients.

A disadvantage of direct methods is that joint optimization of dense structure and motion in real-time is still an open research problem. For this reason, the standard approach is to estimate the latest camera pose with respect to a previously accumulated dense map and subsequently, given a set of estimated camera poses, update the dense map [Newcombe et al., 2011b, Ondruska et al., 2015]. Clearly, this separation of tracking and mapping only results in optimal accuracy when the output of each stage yields the optimal estimate. Other algorithms optimize a graph of poses but do not allow a deformation of the structure once triangulated [Engel et al., 2014]. Contrarily, some algorithms ignore the camera poses and instead allow non-rigid deformation of the 3D structure [Whelan et al., 2014, 2015]. The obtained results are accurate and visually impressive, however, a thorough probabilistic treatment is missing when processing measurements, separating tracking and mapping, or fixating and removing states. To the best of our knowledge, it is therefore currently not possible to obtain accurate covariance estimates from dense VO. Hence, the *consistent fusion* [Bar-Shalom et al., 2001, Kottas et al., 2012] with complementary sensors (*e.g.*, inertial) is currently not possible. In the proposed work, we use direct methods only to establish feature correspondence. Subsequently, bundle adjustment can be used for joint optimization of structure and motion where it is also possible to include inertial measurements as we have demonstrated in previous work [Forster et al., 2015a].

System Overview

Figure B.1 provides an overview of the proposed approach. We use two parallel threads (as in [Klein and Murray, 2007]), one for estimating the camera motion, and a second one for mapping as the environment is being explored. This separation allows fast and constant-time tracking in one thread, while the second thread extends the map, decoupled from hard real-time constraints.

The motion-estimation thread implements the proposed semi-direct approach to motion estimation. Our approach is divided into three steps: sparse image alignment, relaxation, and refinement (Fig. B.1). Sparse image alignment estimates frame-to-frame motion by minimizing the intensity difference of features that correspond to the projected location of the same 3D points. A subsequent step relaxes the geometric con-

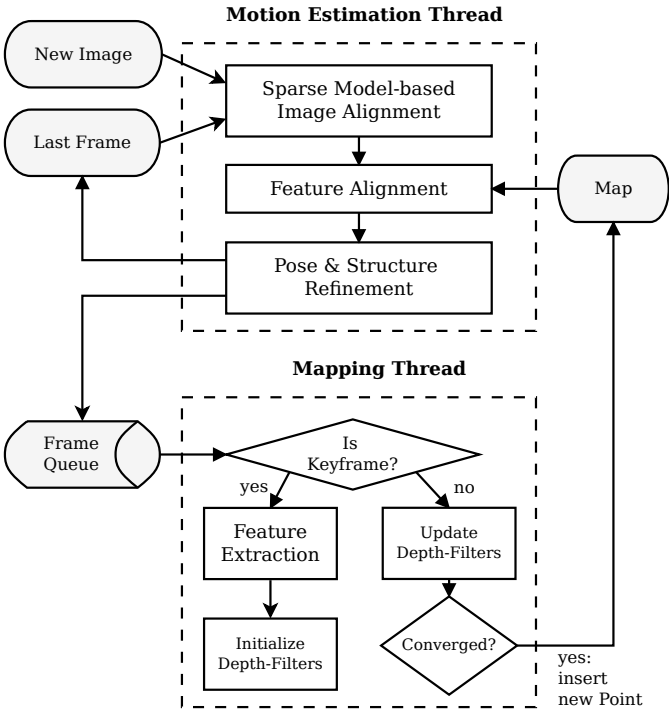


Figure B.1 – Tracking and mapping pipeline

straint to obtain sub-pixel feature correspondence. This step introduces a reprojection error, which we finally refine by means of bundle adjustment.

In the mapping thread, a probabilistic depth-filter is initialized for each feature for which the corresponding 3D point is to be estimated. New depth-filters are initialized whenever a new keyframe is selected for corner pixels as well as for pixels along intensity gradient edges. The filters are initialized with a large uncertainty in depth and undergo a recursive Bayesian update with every subsequent frame. When a depth filter's uncertainty becomes small enough, a new 3D point is inserted in the map and is immediately used for motion estimation.

Notation

The intensity image recorded from a moving camera C at timestep k is denoted with $\mathtt{I}_k^C : \Omega^C \subset \mathbb{R}^2 \mapsto \mathbb{R}$, where Ω^C is the image domain. Any 3D point $\boldsymbol{\rho} \in \mathbb{R}^3$ maps to the image coordinates $\mathbf{u} \in \mathbb{R}^2$ through the camera projection model: $\mathbf{u} = \pi(\boldsymbol{\rho})$. Given the inverse scene depth $\rho > 0$ at pixel $\mathbf{u} \in \mathcal{R}_k^C$, the position of a 3D point is obtained using the back-projection model $\boldsymbol{\rho} = \pi_\rho^{-1}(\mathbf{u})$. Where we denote with $\mathcal{R}_k^C \subseteq \Omega$ those pixels for which the depth is known at time k in camera C. The projection models are known from prior calibration [Furgale et al., 2013].

The position and orientation of the world frame W with respect to the k^{th} camera frame

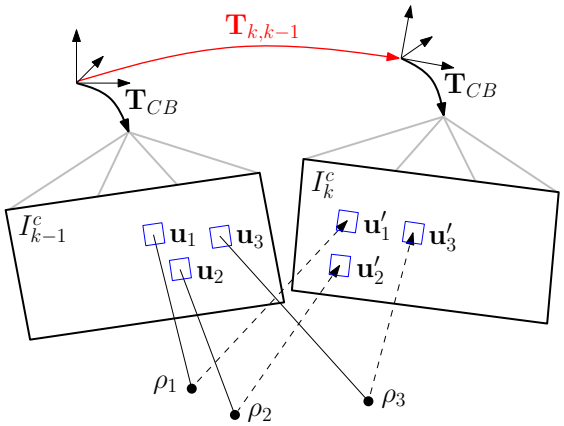


Figure B.2 – Changing the relative pose $\mathrm{T}_{k,k-1}$ between the current and the previous frame implicitly moves the position of the reprojected points in the new image $\mathbf{u}'_i$. Sparse image alignment seeks to find $\mathrm{T}_{k,k-1}$ that minimizes the photometric difference between image patches corresponding to the same 3D point (blue squares). Note, in all figures, the parameters to optimize are drawn in red and the optimization cost is highlighted in blue.

is described by the rigid body transformation $\mathrm{T}_{k\mathrm{W}} \in \mathrm{SE}(3)$ [Ma et al., 2005]. A 3D point ${}_{\mathrm{W}}\boldsymbol{\rho}$ that is expressed in world coordinates can be transformed to the k^{th} camera frame using: ${}_{k}\boldsymbol{\rho} = \mathrm{T}_{k\mathrm{W}}\ {}_{\mathrm{W}}\boldsymbol{\rho}$.

Motion Estimation

In this section, we describe the proposed semi-direct approach to motion estimation, which assumes that the position of some 3D points corresponding to features in previous frames are known from prior depth estimation.

Sparse Image Alignment

Image to model alignment estimates the incremental camera motion by minimizing the intensity difference (*photometric error*) of pixels that observe the same 3D point.

To simplify a later generalization to multiple cameras, we introduce a *body frame* B that is rigidly attached to the camera frame C with known extrinsic calibration $\mathrm{T}_{\mathrm{CB}} \in \mathrm{SE}(3)$ (see Fig. B.2). Our goal is to estimate the incremental motion of the body frame $\mathrm{T}_{kk-1} \doteq \mathrm{T}_{\mathrm{B}_k\mathrm{B}_{k-1}}$ such that the photometric error is minimized:

$$\mathrm{T}^{\star}_{kk-1} = \arg\min_{\mathrm{T}_{kk-1}} \sum_{\mathbf{u}\in\bar{\mathcal{R}}^{\mathrm{C}}_{k-1}} \frac{1}{2} \|\mathbf{r}_{\mathrm{I}^{\mathrm{C}}_{\mathbf{u}}}(\mathrm{T}_{kk-1})\|^2_{\Sigma_{\mathrm{I}}}, \tag{B.1}$$

where the photometric residual $\mathbf{r}_{\mathrm{I}^{\mathrm{C}}_{\mathbf{u}}}$ is defined by the intensity difference of pixels in

subsequent images $\mathtt{I}^{\mathrm{C}}_k$ and $\mathtt{I}^{\mathrm{C}}_{k-1}$ that observe the same 3D point $\boldsymbol{\rho}_{\mathbf{u}}$:

$$\mathbf{r}_{\mathtt{I}^{\mathrm{C}}_{\mathbf{u}}}(\mathtt{T}_{kk-1}) \doteq \mathtt{I}^{\mathrm{C}}_k\Big(\pi(\mathtt{T}_{\mathrm{CB}}\mathtt{T}_{kk-1}\,\boldsymbol{\rho}_{\mathbf{u}})\Big) - \mathtt{I}^{\mathrm{C}}_{k-1}\Big(\pi(\mathtt{T}_{\mathrm{CB}}\,\boldsymbol{\rho}_{\mathbf{u}})\Big). \tag{B.2}$$

The 3D point $\boldsymbol{\rho}_{\mathbf{u}}$ (which is expressed in the reference frame B_{k-1}) can be computed for pixels with known depth by means of back-projection:

$$\boldsymbol{\rho}_{\mathbf{u}} = \mathtt{T}_{\mathrm{BC}}\,\pi^{-1}_{\rho}(\mathbf{u}), \qquad \forall\,\mathbf{u} \in \mathcal{R}^{\mathrm{C}}_{k-1}, \tag{B.3}$$

However, the optimization in Eq. (B.1) includes only a subset of those pixels $\bar{\mathcal{R}}^{\mathrm{C}}_{k-1} \subseteq \mathcal{R}^{\mathrm{C}}_{k-1}$, namely those for which the back-projected points are also visible in the image $\mathtt{I}^{\mathrm{C}}_k$:

$$\bar{\mathcal{R}}^{\mathrm{C}}_{k-1} = \Big\{\mathbf{u} \;\Big|\; \mathbf{u} \in \mathcal{R}^{\mathrm{C}}_{k-1} \;\wedge\; \pi\Big(\mathtt{T}_{\mathrm{CB}}\mathtt{T}_{kk-1}\mathtt{T}_{\mathrm{BC}}\,\pi^{-1}_{\rho}(\mathbf{u})\Big) \in \Omega^{\mathrm{C}}\Big\}.$$

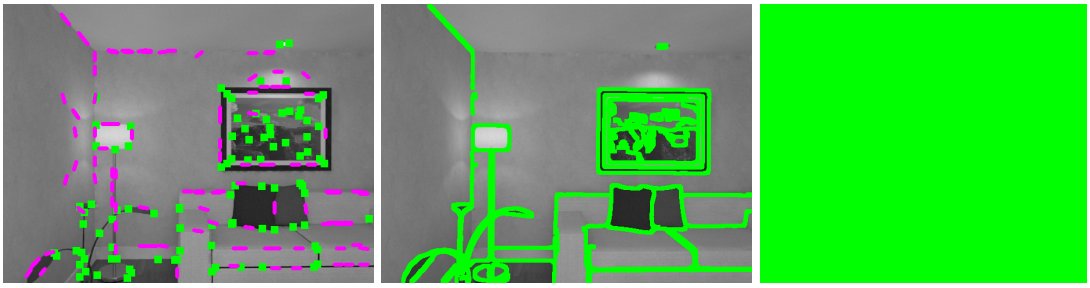
Image to model alignment has previously been used in the literature to estimate camera motion. Apart from minor variations in the formulation, the main difference among the approaches is the source of the depth information as well as the region $\mathcal{R}^{\mathrm{C}}_{k-1}$ in image $\mathtt{I}^{\mathrm{C}}_k$ for which the depth is known. As discussed in Section B.2, we denote methods that know and exploit the depth for all pixels in the reference view as *dense* methods [Newcombe et al., 2011b]. Converseley, approaches that only perform the alignment for pixels with high image gradients are denoted *semi-dense* [Engel et al., 2013]. In this paper, we propose a novel *sparse* image alignment approach that assumes known depth only for corners and features lying on intensity edges. Fig. B.3 summarizes our notation of dense, semi-dense, and sparse approaches.

To make the sparse approach more robust, we propose to aggregate the photometric cost in a small patch centered at the feature pixel. Since the depth for neighboring pixels is unknown, we approximate it with the same depth that was estimated for the feature.

To summarize, sparse image alignment solves the non-linear least squares problem in Eq. (B.1) with $\mathcal{R}^{\mathrm{C}}_{k-1}$ corresponding to small patches centered at corner and edgelet features with known depth. This optimization can be solved efficiently using standard iterative non-linear least squares algorithms such as Levenberg-Marquardt. More details on the optimization, including the analytic Jacobians, are provided in the Appendix.

Relaxation and Refinement

Sparse image alignment is an efficient method to estimate the incremental motion between subsequent frames. However, to minimize drift in the motion estimate, it is paramount to register a new frame to the oldest frame possible. One approach is to



(a) Sparse (b) Semi-Dense (c) Dense

Figure B.3 – An image from the *ICL-NUIM* dataset (Sec. B.11.2) with pixels used for image-to-model alignment (marked in green for corners and magenta for edgelets) for sparse, semi-dense, and dense methods. Dense approaches (c) use every pixel in the image, semi-dense (b) use just the pixels with high intensity gradient, and the proposed sparse approach (a) uses selected pixels at corners or along intensity gradient edges.

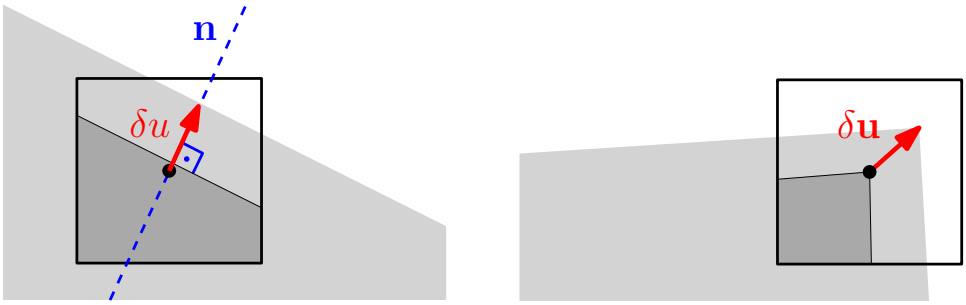
use an older frame as reference for image alignment [Engel et al., 2014]. However, the robustness of the alignment cannot be guaranteed as the distance between the frames in the alignment increases (see experiment in Section B.11.1). We therefore propose to relax the geometric constraints given by the reprojection of 3D points and to perform an individual 2D alignment of corresponding feature patches. The alignment of each patch in the new frame is performed with respect to a reference patch from the frame where the feature was first extracted; hence, the oldest frame possible, which should maximally minimize feature drift. However, the 2D alignment generates a reprojection error that is the difference between the projected 3D point and the aligned feature position. Therefore, in a final step, we perform bundle adjustment to optimize both the 3D point's position and the camera poses such that this reprojection error is minimized.

In the following, we detail our approach to feature alignment and bundle adjustment. Thereby, we take special care of features lying on intensity gradient edges.

2D feature alignment minimizes the intensity difference of a small image patch $\mathcal{P}$ that is centered at the projected feature position $\mathbf{u}'$ in the newest frame k with respect to a reference patch from the frame r where the feature was first observed (see Fig. B.4). To improve the accuracy of the alignment, we apply an affine warping $\mathbf{A}$ to the reference patch, which is computed from the estimated relative pose $\mathtt{T}_{kr}$ between the reference frame and the current frame [Klein and Murray, 2007]. For corner features, the optimization computes a correction $\delta\mathbf{u}^\star \in \mathbb{R}^2$ to the predicted feature position $\mathbf{u}'$ that minimizes the photometric cost:

$$\mathbf{u}'^\star = \mathbf{u}' + \delta\mathbf{u}^\star, \quad \text{with} \quad \mathbf{u}' = \pi\Big(\mathtt{T}_{\mathrm{CB}}\,\mathtt{T}_{kr}\,\mathtt{T}_{\mathrm{BC}}\,\pi_\rho^{-1}(\mathbf{u})\Big) \tag{B.4}$$

$$\delta\mathbf{u}^\star = \arg\min_{\delta\mathbf{u}} \sum_{\Delta\mathbf{u}\in\mathcal{P}} \frac{1}{2}\Big\| \mathtt{I}_k^{\mathrm{C}}\left(\mathbf{u}'+\delta\mathbf{u}+\Delta\mathbf{u}\right) - \mathtt{I}_r^{\mathrm{C}}(\mathbf{u}+\mathbf{A}\Delta\mathbf{u})\Big\|^2,$$



(a) Edge alignment. (b) Corner alignment.

Figure B.4 – Different alignment strategies for corners and edgelets. The alignment of an edge feature is restricted to the normal direction **n** of the edge.

where $\Delta\mathbf{u}$ is the iterator variable that is used to compute the sum over the patch $\mathcal{P}$. This alignment is solved using the inverse compositional Lucas-Kanade algorithm [Baker and Matthews, 2004].

For features lying on intensity gradient edges, 2D feature alignment is problematic because of the aperture problem — features may drift along the edge. Therefore, we limit the degrees of freedom in the alignment to the normal direction to the edge. This is illustrated in Fig. B.4a, where a warped reference feature patch is schematically drawn at the predicted position in the newest image. For features on edges, we therefore optimize for a scalar correction $\delta u^\star \in \mathbb{R}$ in the direction of the edge normal $\mathbf{n}$ to obtain the corresponding feature position $\mathbf{u}'^\star$ in the newest frame:

$$\mathbf{u}'^\star = \mathbf{u}' + \delta u^\star \cdot \mathbf{n}, \quad \text{with} \tag{B.5}$$
$$\delta u^\star = \arg\min_{\delta u} \sum_{\Delta\mathbf{u}\in\mathcal{P}} \frac{1}{2} \left\| \mathtt{I}_k^\mathtt{C}\left(\mathbf{u}' + \delta u \cdot \mathbf{n} + \Delta\mathbf{u}\right) - \mathtt{I}_r^\mathtt{C}(\mathbf{u} + \mathbf{A}\Delta\mathbf{u}) \right\|^2.$$

This is similar to previous work on VO with edgelets, where feature correspondence is found by sampling along the normal direction for abrupt intensity changes [Harris and Stennett, 1990, Drummond and Cipolla, 2002, Comport et al., 2003, Vacchetti et al., 2004, Reitmayr and Drummond, 2006, Klein and Murray, 2008]. However, in our case, sparse image alignment provides a very good initialization of the feature position, which directly allows us to follow the intensity gradient in an optimization.

After feature alignment, we have established feature correspondence with subpixel accuracy. However, feature alignment violated the epipolar constraints and introduced a reprojection error $\delta\mathbf{u}$, which is typically well below 0.5 pixels. Therefore, in the last step of motion estimation, we refine the camera poses and landmark positions

$\mathcal{X} = \{\mathrm{T}_{kW}, \boldsymbol{\rho}_i\}$ by minimizing the squared sum of reprojection errors:

$$\mathcal{X}^\star = \arg\min_{\mathcal{X}} \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{L}_k^C} \frac{1}{2} \|\mathbf{u}'^\star_i - \pi\Big(\mathrm{T}_{CB}\ \mathrm{T}_{kW}\ \boldsymbol{\rho}_i\Big)\|^2 + \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{L}_k^E} \frac{1}{2} \|\mathbf{n}_i^\mathsf{T}\Big(\mathbf{u}'^\star_i - \pi\Big(\mathrm{T}_{CB}\ \mathrm{T}_{kW}\ \boldsymbol{\rho}_i\Big)\Big)\|^2 \tag{B.6}$$

where $\mathcal{K}$ is the set of all keyframes in the map, $\mathcal{L}_k^C$ the set of all landmarks corresponding to corner features, and $\mathcal{L}_k^E$ the set of all edge features that were observed in the k^{th} camera frame. The reprojection error of edge features is projected along the edge normal because the component along the edge cannot be determined.

The optimization problem in Eq. (B.6) is a standard bundle adjustment problem that can be solved in real-time using iSAM2 [Kaess et al., 2012]. In [Forster et al., 2015a] we further show how the objective function can be extended to include inertial measurements.

While optimization over the whole trajectory in Eq. (B.6) results in the most accurate results (see Sec. B.11.2), we found that for many applications (*e.g.* for state estimation of micro aerial vehicles [Forster et al., 2014b, Faessler et al., 2015]) it suffices to only optimize the latest camera pose and the 3D points separately.

Mapping

In the previous section, we assumed that the depth at sparse feature locations in the image is known. In this section, we describe how the mapping thread estimates this depth for newly detected features. Therefore, we assume that the camera poses are known from the motion estimation thread.

The depth at a single pixel is estimated from multiple observations by means of a recursive Bayesian *depth filter*. New depth filters are initialized at intensity corners and along gradient edges when the number of tracked features falls below some threshold and, therefore, a keyframe is selected. Every depth filter is associated to a reference keyframe r, where the initial depth uncertainty is initialized with a large value. For a set of previous keyframes[3] as well as every subsequent frame with known relative pose $\{\mathrm{I}_k, \mathrm{T}_{kr}\}$, we search for a patch along the epipolar line that has the highest correlation (see Fig. B.5). Therefore, we move the reference patch along the epipolar line and compute the zero mean sum of squared differences. From the pixel with maximum

[3] In the previous publication of SVO [Forster et al., 2014b] and in the open source implementation we suggested to update the depth filter only with newer frames $k > r$, which works well for down-looking cameras in micro aerial vehicle applications. However, for forward motions, it is beneficial to update the depth filters also with previous frames $k < r$, which increases the performance with forward-facing cameras.

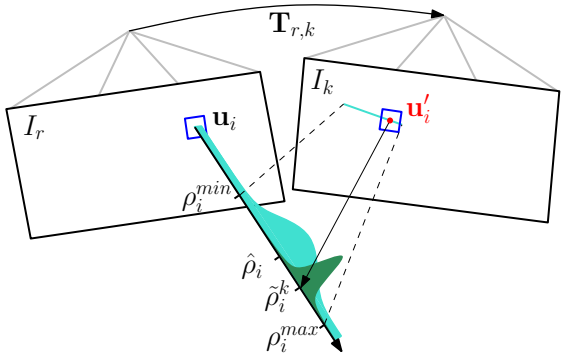


Figure B.5 – Probabilistic depth estimate $\hat{\rho}_i$ for feature i in the reference frame r. The point at the true depth projects to similar image regions in both images (blue squares). Thus, the depth estimate is updated with the triangulated depth $\tilde{\rho}_i^k$ computed from the point $\mathbf{u}'_i$ of highest correlation with the reference patch. The point of highest correlation lies always on the epipolar line in the new image.

correlation, we triangulate the depth measurement $\tilde{\rho}_i^k$, which is used to update the depth filter. If enough measurements were obtained such that uncertainty in the depth is below a certain threshold, we initialize a new 3D point at the estimated depth, which subsequently can be used for motion estimation (see system overview in Fig. B.1). This approach for depth estimation also works for features on gradient edges. Due to the aperture problem, we however skip measurements where the edge is parallel to the epipolar line.

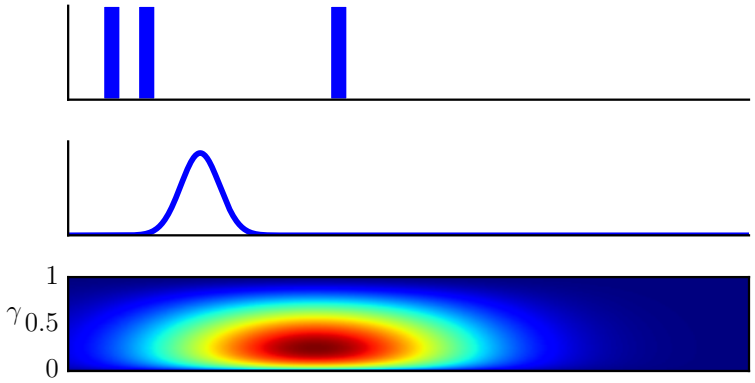
Ideally, we would like to model the depth with a non-parametric distribution to deal with multiple depth hypotheses (top rows in Fig. B.6). However, this is computationally too expensive. Therefore, we model the depth filter according to [Vogiatzis and Hernández, 2011] with a two dimensional distribution: the first dimension is the inverse depth ρ [Civera et al., 2008], while the second dimension γ is the inlier probability (see bottom rows in Fig. B.6). Hence, a measurement $\tilde{\rho}_i^k$ is modeled with a *Gaussian + Uniform* mixture model distribution: an inlier measurement is normally distributed around the true inverse depth ρ_i while an outlier measurement arises from a uniform distribution in the interval $[\rho_i^{\min}, \rho_i^{\max}]$:

$$p(\tilde{\rho}_i^k|\rho_i,\gamma_i) = \gamma_i \mathcal{N}\left(\tilde{\rho}_i^k \middle| \rho_i, \tau_i^2\right) + (1-\gamma_i)\mathcal{U}\left(\tilde{\rho}_i^k \middle| \rho_i^{\min}, \rho_i^{\max}\right), \tag{B.7}$$

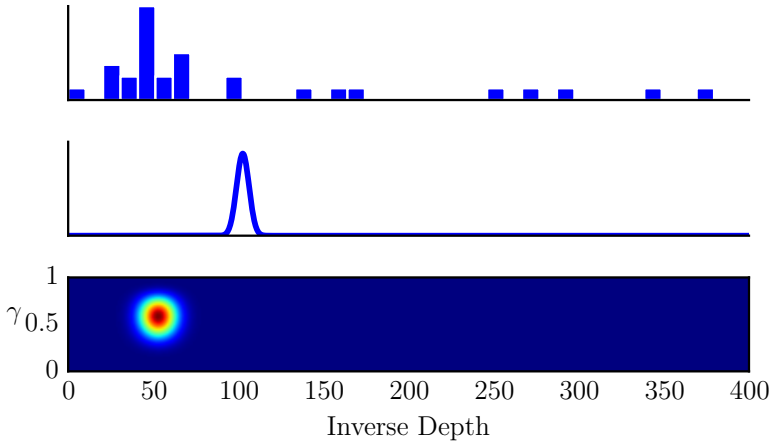
where τ_i^2 the variance of a good measurement that can be computed geometrically by assuming a disparity variance of one pixel in the image plane [Pizzoli et al., 2014].

Assuming independent observations, the Bayesian estimation for ρ on the basis of the measurements $\tilde{\rho}_{r+1}, \ldots, \tilde{\rho}_k$ is given by the posterior

$$p(\rho,\gamma|\tilde{\rho}_{r+1},\ldots,\tilde{\rho}_k) \propto p(\rho,\gamma)\prod_k p(\tilde{\rho}_k|\rho,\gamma), \tag{B.8}$$



(a) After 3 measurements with 70% inlier probability.



(b) After 30 measurements with 70% inlier probability.

Figure B.6 – Illustration of posterior distributions for depth estimation. The histogram in the top rows show the measurements affected by outliers. The distribution in the middle rows show the posterior distribution when modeling the depth with a single variate Gaussian distribution. The bottom rows show the posterior distribution of the proposed approach that is using the model from [Vogiatzis and Hernández, 2011]. The distribution is bi-variate and models the inlier probability (vertical axis) together with the inverse depth (horizontal axis).

with $p(\rho, \gamma)$ being a prior on the true inverse depth and the ratio of good measurements supporting it. For incremental computation of the posterior, the authors of [Vogiatzis and Hernández, 2011] show that (B.8) can be approximated by the product of a Gaussian distribution for the depth and a Beta distribution for the inlier ratio:

$$q(\rho, \gamma | a_k, b_k, \mu_k, \sigma_k^2) = Beta(\gamma | a_k, b_k) \mathcal{N}(\rho | \mu_k, \sigma_k^2), \tag{B.9}$$

where a_k and b_k are the parameters controlling the Beta distribution. The choice is motivated by the fact that the $Beta \times Gaussian$ is the approximating distribution minimizing the Kullback-Leibler divergence from the true posterior (B.8). Upon the

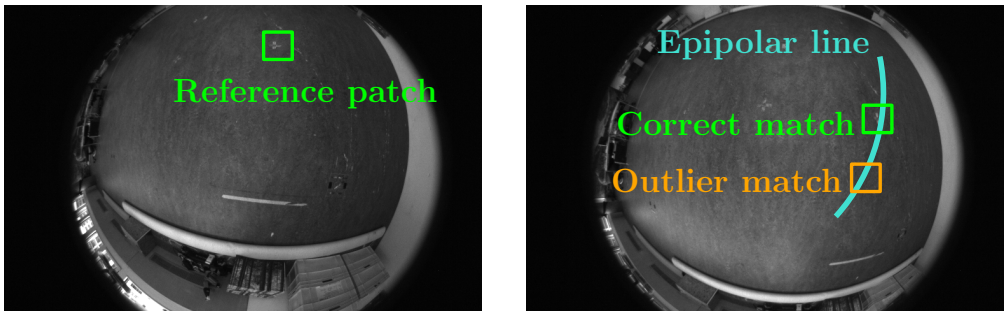


Figure B.7 – Illustration of the epipolar search to estimate the depth of the pixel in the center of the reference patch in the left image. Given the extrinsic and intrinsic calibration of the two images, the epipolar line that corresponds to the reference pixel is computed. Due to self-similar texture, erroneous matches along the epipolar line are frequent.

k-th observation, the update takes the form

$$p(\rho, \gamma | \tilde{\rho}_{r+1}, \ldots, \tilde{\rho}_k) \approx q(d, \gamma | a_{k-1}, b_{k-1}, \mu_{k-1}, \sigma^2_{k-1}) \cdot p(\tilde{\rho}_k | d, \gamma) \cdot \text{const}, \tag{B.10}$$

and the authors of [Vogiatzis and Hernández, 2011] approximated the true posterior (B.10) with a *Beta* $\times$ *Gaussian* distribution by matching the first and second order moments for $\hat{d}$ and γ. The updates formulas for a_k, b_k, μ_k and σ^2_k are thus derived and we refer to the original work in [Vogiatzis and Hernández, 2011] for the details on the derivation.

Fig. B.6 shows a small simulation experiment that highlights the advantage of the model proposed in [Vogiatzis and Hernández, 2011]. The histogram in the top rows show the measurements that are corrupted by 30% outlier measurements. The distribution in the middle rows show the posterior distribution when modeling the depth with a single variate Gaussian distribution as used for instance in [Engel et al., 2013]. Outlier measurements have a huge influence on the mean of the estimate. The figures in the bottom rows show the posterior distribution of the proposed approach that is using the model from [Vogiatzis and Hernández, 2011] with the inlier probability drawn in the vertical axis. As more measurements are received at the same depth, the inlier probability increases. In this model, the mean of the estimate is less affected by outliers while the inlier probability is informative about the confidence of the estimate. Fig. B.7 shows qualitatively the importance of robust depth estimation in self-similar environments, where outlier matches are frequent.

In [Pizzoli et al., 2014] we demonstrate how the same depth filter can be used for *dense* mapping.

(a) Perspective (b) Fisheye (c) Catadioptric

Figure B.8 – Different optical distortion models that are supported by SVO.

Large Field of View Cameras

To model large optical distortion, such as fisheye and catadioptric (see Fig. B.8), we use the camera model proposed in [Scaramuzza et al., 2006], which models the projection $\pi(\cdot)$ and unprojection $\pi^{-1}(\cdot)$ functions with polynomials. Using the Jacobians of the camera distortion in the sparse image alignment and bundle adjustment step is sufficient to enable motion estimation for large FoV cameras.

For estimating the depth of new features (*c.f.*, Sec. B.6), we need to sample pixels along the epipolar line. For distorted images, the epipolar line is curved (see Fig. B.7). Therefore, we regularly sample the *great circle*, which is the intersection of the epipolar plane with the unit sphere centered at the camera pose of interest. The angular resolution of the sampling corresponds approximately to one pixel in the image plane. For each sample, we apply the camera projection model $\pi(\cdot)$ to obtain the corresponding pixel coordinate on the curved epipolar line.

Multi-Camera Systems

The proposed semi-direct camera motion estimation starts directly with an optimization of the relative pose T_{kk-1}. Since in Sec. B.5.1 we already introduced a body frame B that is rigidly attached to the camera, it is now straightforward to generalize sparse image alignment to multiple rigidly attached and synchronized cameras. Let us assume that given a camera rig with M cameras (see Fig. B.9). The extrinsic calibration of the individual cameras $c \in \mathrm{C}$ with respect to the body frame T_{CB} is assumed to be known

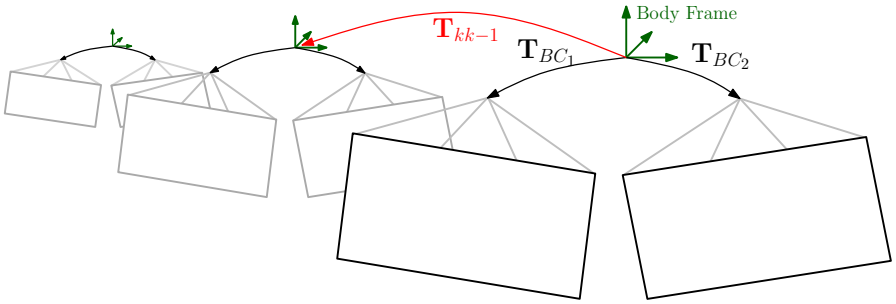


Figure B.9 – Visual odometry with multiple rigidly attached and synchronized cameras. We know the relative pose of each camera to the body frame $\mathrm{T}_{\mathrm{BC}_j}$ from extrinsic calibration and the goal is to estimate the relative motion of the body frame T_{kk-1}.

from prior extrinsic calibration[4]. To use multiple cameras, we only need to add an extra summation in the cost function of Eq. (B.1) to use the information from all images for sparse image alignment:

$$\mathrm{T}^{\star}_{kk-1} = \arg\min_{\mathrm{T}_{kk-1}} \sum_{\mathrm{C}\in\mathbf{C}} \sum_{\mathbf{u}\in\bar{\mathcal{R}}^{\mathrm{C}}_{k-1}} \frac{1}{2} \| \mathbf{r}_{\mathrm{I}^{\mathrm{C}}_{\mathbf{u}}}(\mathrm{T}_{kk-1}) \|^2_{\Sigma_{\mathrm{I}}}. \tag{B.11}$$

The same summation is necessary in the bundle adjustment step to sum the reprojection errors from all cameras. The remaining steps of feature alignment and mapping are independent of how many cameras are used. To summarize, the only modification to enable the use of multiple cameras is to refer the optimizations to a central body frame, which requires us to include the extrinsic calibration T_{CB} in the Jacobians as shown in the Appendix.

Motion Priors

In feature-poor environments, during rapid motions, or in case of dynamic obstacles it can be very helpful to employ a motion prior. A motion prior is an additional term that is added to the cost function in Eq. (B.11), which penalizes motions that are not in agreement with the prior estimate. Thereby, "jumps" in the motion estimate due to unconstrained degrees of freedom or outliers can be suppressed. In a car scenario for instance, a constant velocity motion model may be assumed as the inertia of the car prohibits sudden changes from one frame to the next. Other priors may come from additional sensors such as gyroscopes, which allow us to measure the incremental rotation between two frames.

Let us assume that we are given a relative translation prior $\tilde{\mathbf{p}}_{kk-1}$ (*e.g.*, from a constant velocity assumption) and a relative rotation prior $\tilde{\mathrm{R}}_{kk-1}$ (*e.g.*, from integrating a gyro-

[4]We use the calibration toolbox *Kalibr* [Furgale et al., 2013], which is available at https://github.com/ethz-asl/kalibr

scope). In this case, we can employ a motion prior by adding additional terms to the cost of the sparse image alignment step:

$$\begin{aligned} \mathrm{T}_{kk-1}^{\star} = \arg\min_{\mathrm{T}_{kk-1}} \sum_{C\in\mathrm{C}} \sum_{\mathbf{u}\in\mathcal{R}_{k-1}^{C}} \frac{1}{2}\|\mathbf{r}_{\mathrm{I}_{\mathbf{u}}^{C}}(\mathrm{T}_{kk-1})\|_{\Sigma_{\mathrm{I}}}^{2} \\ + \frac{1}{2}\|\mathbf{p}_{kk-1} - \tilde{\mathbf{p}}_{kk-1}\|_{\Sigma_{\mathbf{p}}}^{2} \\ + \frac{1}{2}\|\log(\tilde{\mathrm{R}}_{kk-1}^{\mathsf{T}}\mathrm{R}_{kk-1})^{\vee}\|_{\Sigma_{\mathrm{R}}}^{2}, \end{aligned} \tag{B.12}$$

where the covariances $\Sigma_{\mathbf{p}}, \Sigma_{\mathrm{R}}$ are set according to the uncertainty of the motion prior and the variables $(\mathbf{p}_{kk-1}, \mathrm{R}_{kk-1}) \doteq \mathrm{T}_{kk-1}$ are the current estimate of the relative position and orientation (expressed in body coordinates B). The *logarithm map* maps a rotation matrix to its rotation vector (see Eq. (B.18)). Note that the same cost function can be added to the bundle adjustment step. For further details on solving Eq. (B.12), we refer the interested reader to the Appendix.

Implementation Details

In this section we provide additional details on various aspects of our implementation.

Initialization

The algorithm is bootstrapped to obtain the pose of the first two keyframes and the initial map using the 5-point relative pose algorithm from [Nister, 2004].

Sparse Image Alignment

For sparse image alignment, we use a patch size of 4×4 pixels. In the experimental section we demonstrate that the sparse approach with such a small patch size achieves comparable performance to semi-dense and dense methods in terms of robustness when the inter-frame distance is small, which typically is true for frame-to-frame motion estimation. In order to cope with large motions, we apply the sparse image alignment algorithm in a coarse-to-fine scheme. Therefore, the image is halfsampled to create an image pyramid of five levels. The photometric cost is then optimized at the coarsest level until convergence, starting from the initial condition $\mathrm{T}_{kk-1} = \mathbf{I}_{4\times4}$. Subsequently, the optimization is continued at the next finer level to improve the precision of the result. To save processing time, we stop after convergence on the third level, at which stage the estimate is accurate enough to initialize feature alignment. To increase the robustness against dynamic obstacles, occlusions and reflections, we additionally employ a robust cost function [Kerl et al., 2013, MacTavish and Barfoot,

2015].

Feature Alignment

For feature alignment we use a patch-size of 8×8 pixels. Since the reference patch may be multiple frames old, we employ an affine illumination model to cope with illumination changes [Jin et al., 2001]. For all experiments we limit the number of matched features to 180 in order to guarantee a constant cost per frame.

Mapping

In the mapping thread, we divide the image in cells of fixed size (e.g., 32×32 pixels). For every keyframe a new depth-filter is initialized at the FAST corner [Rosten et al., 2010] with highest score in the cell unless there is already a 2D-to-3D correspondence present. In cells where no corner is found, we detect the pixel with highest gradient magnitude and initialize an edge feature. This results in evenly distributed features in the image.

To speed up the depth-estimation we only sample a short range along the epipolar line; in our case, the range corresponds to twice the standard deviation of the current depth estimate. We use a 8×8 pixel patch size for the epipolar search.

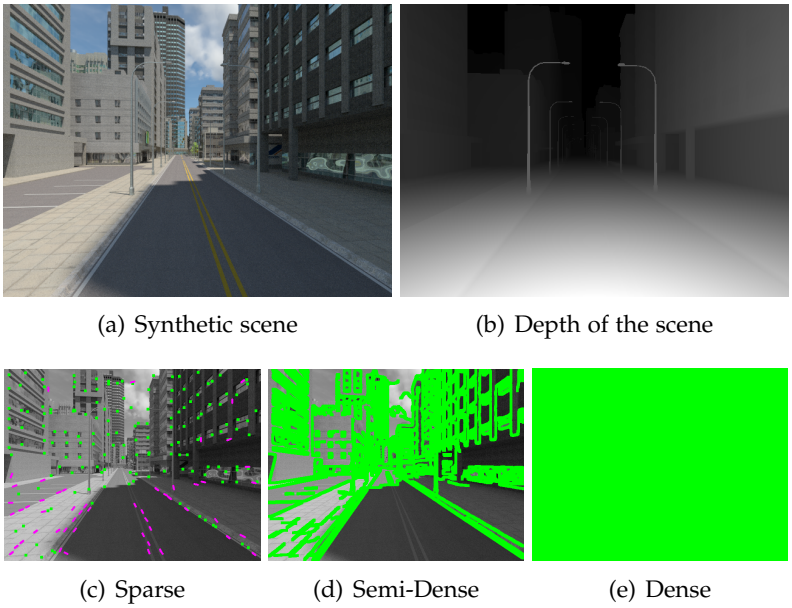
Experimental Evaluation

We implemented the proposed VO system in C++ and tested its performance in terms of accuracy, robustness, and computational efficiency. We first compare the proposed sparse image alignment algorithm against semi-dense and dense image alignment algorithms and investigate the influence of the patch size used in the sparse approach. Finally, in Sec. B.11.2 we compare the full pipeline in different configurations against the state of the art on nine different dataset sequences.

Image Alignment: From Sparse to Dense

In this section we evaluate the robustness of the proposed sparse image alignment algorithm (Sec. B.5.1) and compare its performance to semi-dense and dense image alignment alternatives. Additionally, we investigate the influence of the patch-size that is used for the sparse approach.

The experiment is based on a synthetic dataset with known camera motion, depth



(a) Synthetic scene (b) Depth of the scene

(c) Sparse (d) Semi-Dense (e) Dense

Figure B.10 – An image from the *Urban Canyon* dataset [Zhang et al., 2016] (Sec. B.11.1) with pixels used for image-to-model alignment (marked in green) for sparse, semi-dense, and dense methods. Dense approaches use every pixel in the image, semi-dense use just the pixels with high intensity gradient, and the proposed sparse approach uses selected pixels at corners or along intensity gradient edges.

and calibration [Zhang et al., 2016].[5] The camera performs a forward motion through an urban canyon as the excerpt of the dataset in Fig. B.10a shows. The dataset consists of 2500 frames with 0.2 meters distance between frames and a median scene depth of 12.4 meters. For the experiment, we select a reference image $\mathtt{I}_r$ with known depth (see Fig. B.10b) and estimate the relative pose $\mathtt{T}_{rk}$ of 60 subsequent images $k \in \{r+1, \ldots, r+60\}$ along the trajectory by means of image to model alignment. For each image pair $\{\mathtt{I}_r, \mathtt{I}_k\}$, the alignment is repeated 800 times with initial perturbation that is sampled uniformly within a 2 m range around the true value. We perform the experiment at 18 reference frames along the trajectory. The alignment is considered converged when the estimated relative pose is closer than 0.1 meters from the ground-truth. The goal of this experiment is to study the magnitude of the perturbation from which image to model alignment is capable to converge as a function of the distance to the reference image. The performance in this experiment is a measure of robustness: successful pose estimation from large initial perturbations shows that the algorithm is capable of dealing with rapid camera motions. Furthermore, large distances between the reference image $\mathtt{I}_r$ and test image $\mathtt{I}_k$ simulates the performance at low camera frame-rates.

For the sparse image alignment algorithm, we extract 100 FAST corners in the reference image (see Fig. B.10c) and initialize the corresponding 3D points using the known

[5] The *Urban Canyon* dataset Zhang et al. [2016] is available at http://rpg.ifi.uzh.ch/fov.html

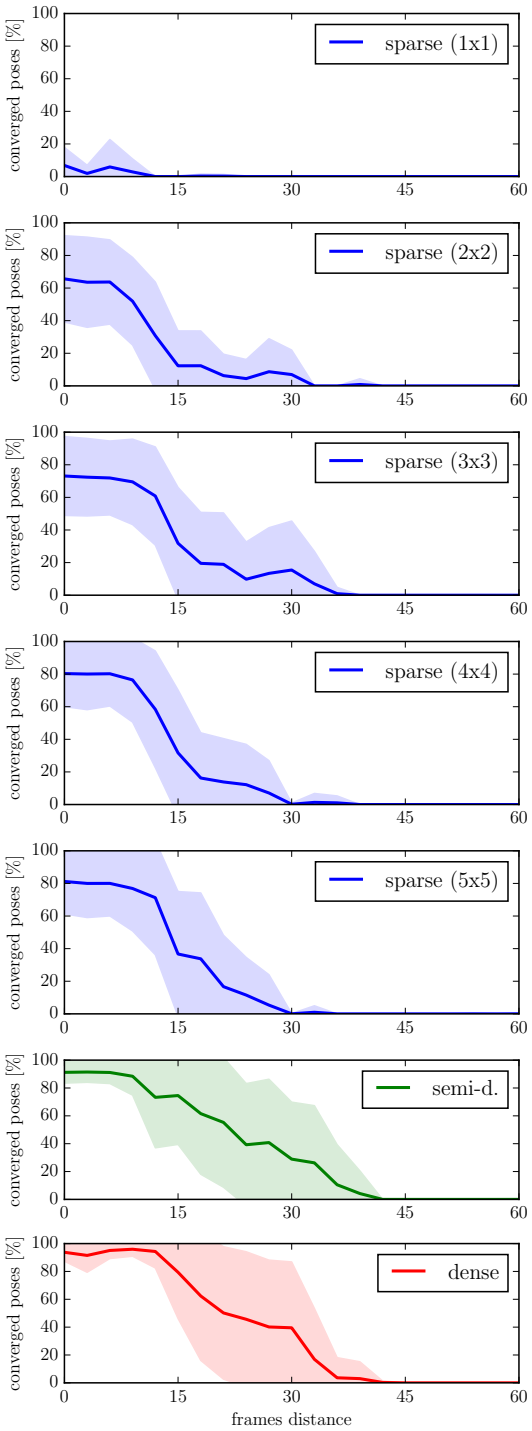
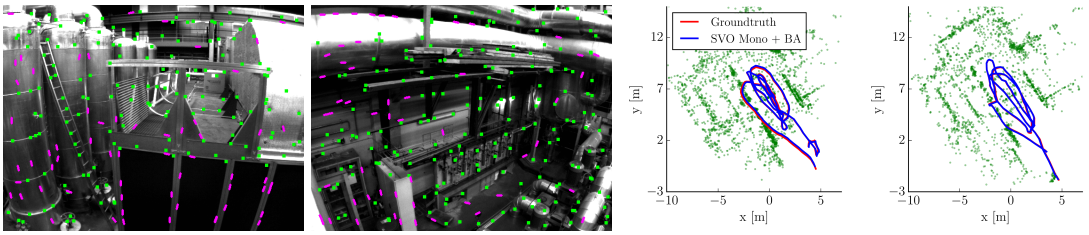


Figure B.11 – Convergence probability of the model-based image alignment algorithm as a function of the distance to the reference image and evaluated for sparse image alignment with patch sizes ranging from 1×1 to 5×5 pixels, semi-dense, and dense image alignment. The colored region highlights the 68% confidence interval.



(a) (b) (c) Machine Hall 1 (d) Machine Hall 2

Figure B.12 – Figures (a) and (b) show excerpts of the EuRoC dataset Burri et al. [2015] with tracked corners marked in green and edgelets marked in magenta. Figures (c) and (d) show the reconstructed trajectory and pointcloud on the first two trajectories of the dataset.

depth-map from the rendering process. We repeat the experiment with patch-sizes ranging from 1×1 pixels to 5×5 pixels. We evaluate the semi-direct approach (as proposed in the LSD framework [Engel et al., 2013]) by using pixels along intensity gradients (see Fig. B.10d). Finally, we perform the experiment using all pixels in the reference image as proposed in DTAM [Newcombe et al., 2011b].

The results of the experiment are shown in Fig. B.11. Each plot shows a variant of the image alignment algorithm with the vertical axis indicating the percentage of converged trials and the horizontal axis the frame index counted from the reference frame. We can observe that the difference between semi-dense image alignment and dense image alignment is marginal. This is because pixels that exhibit no intensity gradient are not informative for the optimization as their Jacobians are zero [Dellaert and Collins, 1999]. We suspect that using all pixels becomes only useful when considering motion blur and image defocus, which is out of the scope of this evaluation. In terms of sparse image alignment, we observe a gradual improvement when increasing the patch size to 4×4 pixels. A further increase of the patch size does not show improved convergence and will eventually suffer from the approximations adopted by not warping the patches according to the surface orientation.

Compared to the semi-dense approach, the sparse approaches do not reach the same convergence radius, particularly in terms of distance to the reference image. For this reason SVO uses sparse image alignment only to align with respect to the previous image (*i.e.*, $k = r + 1$), in contrast to LSD [Engel et al., 2013] which aligns with respect to the last keyframe.

In terms of computational efficiency, we note that the complexity scales linearly with the number of pixels used in the optimization. The plots show that we can trade-off using a high frame rate camera and a sparse approach with a lower frame-rate camera and a semi-dense approach. The evaluation of this trade-off would ideally incorporate the power consumption of both the camera and processors, which is out of the scope of this evaluation.

Real and Synthetic Experiments

In this section, we compare the proposed algorithm against the state of the art on real and synthetic datasets. Therefore, we present results of the proposed pipeline on the EUROC benchmark [Burri et al., 2015], the TUM RGB-D benchmark dataset [Sturm et al., 2012], the synthetic ICL-NUIM dataset [Handa et al., 2014], and our own dataset that compares different field of view cameras. A selection of these experiments, among others (*e.g.*, from the KITTI benchmark), can also be viewed in the video attachment of this paper.

Euroc Datasets

The EUROC dataset [Burri et al., 2015] consists of stereo images and inertial data that was recorded using a VI-Sensor [Nikolic et al., 2014] that was mounted on a micro aerial vehicle and flown inside a machine hall. Extracts from the dataset are shown in Fig. B.12a and B.12b. The dataset provides a precise ground-truth trajectory that was obtained using a Leica MS50 laser tracking system. In Table B.1 we present results of various monocular and stereo configurations of the proposed algorithm on the first two trajectories of the dataset. The trajectories are 65 and 58 meters long respectively.

We compare our algorithm against the open-source versions of ORB-SLAM [Mur-Artal et al., 2015a] and LSD-SLAM [Engel et al., 2014] on these datasets with their default parameter settings. In contrast to SVO, ORB-SLAM and LSD-SLAM detect loop-closures and subsequently perform a global optimization. Hence, during loop-closure refinements, the latest camera pose may undergo significant "jumps". For this reason, we base the evaluation on the final poses of all keyframes in the trajectory. We observed that the initial poses of LSD-SLAM are not accurate due to the initialization procedure and therefore discard the first 350 frames in the evaluation for LSD-SLAM.

To understand the influence of the proposed extensions of SVO, we run the algorithm in various configurations. "SVO Mono" (only corners) uses only the images that were recorded with the left camera of the sensor. "SVO Mono + Prior" indicates that we use measurements from the gyroscope as priors in the image alignment step as we discussed in Sec. B.9. In the next setting we additionally use edgelet features in combination with corner features. In these first three settings, we only optimize the latest pose; conversely, the keyword "Bundle Adjustment" indicates that results were obtained by optimizing the whole history of keyframes by means of the incremental smoothing algorithm iSAM2 [Kaess et al., 2012]. Therefore, we insert and optimize every new keyframe in the iSAM2 graph when a new keyframe is selected. In this setting, we do neither use motion priors nor edgelets. Since SVO is a visual odometry, it does not not detect loop-closures and only maintains a small local map of the last keyframes. To provide a fair comparison with ORB-SLAM and LSD-SLAM, we deactivated their capability to

	Machine Hall 1			Machine Hall 2		
	Mean	Median	RMSE [m]	Mean	Median	RMSE [m]
SVO Mono	0.224	0.198	0.269	0.531	0.356	0.652
SVO Mono + Prior	0.199	0.131	0.270	0.345	0.314	0.408
SVO Mono + Prior + Edgelet	0.171	0.149	0.201	0.368	0.318	0.425
SVO Mono + Bundle Adjustment	**0.048**	**0.042**	**0.057**	**0.061**	**0.060**	**0.072**
SVO Stereo	0.096	0.092	0.104	0.064	0.063	0.070
SVO Stereo + Prior	0.071	0.066	0.078	0.067	0.059	0.072
SVO Stereo + Prior + Edgelet	0.072	0.060	0.083	0.072	0.062	0.077
SVO Stereo + Bundle Adjustment	**0.039**	**0.037**	**0.043**	**0.046**	**0.042**	**0.053**
ORB Mono SLAM (No loop closure)	0.105	0.126	0.114	0.175	0.209	0.190
LSD Mono SLAM (No loop closure)	0.111	0.107	0.125	0.388	0.357	0.428

	Timing		
	Mean [ms]	St.D.	CPU@20 fps
SVO Mono	2.53	0.42	55 ±10 %
SVO Mono + Prior	**2.32**	**0.40**	**70 ± 8** %
SVO Mono + Prior + Edgelet	2.51	0.52	73 ± 7 %
SVO Mono + Bundle Adjustment	5.25	10.89	72 ±13 %
SVO Stereo	4.70	1.31	90 ± 6 %
SVO Stereo + Prior	**3.86**	**0.86**	**90 ± 7** %
SVO Stereo + Prior + Edgelet	4.12	1.11	91 ± 7 %
SVO Stereo + Bundle Adjustment	7.61	19.03	96 ±13 %
ORB Mono SLAM (No loop closure)	29.81	5.67	187 ±32 %
LSD Mono SLAM (No loop closure)	23.23	5.87	236 ±37 %

Table B.1 – Absolute translation errors in meters after 6 DoF alignment with the ground-truth trajectory and timing measurements on laptop computer with Intel Core i7-4810MQ CPU (2.80 GHz) averaged over three runs of the EUROC Machine Hall 01 dataset. Loop closure detection and optimization was deactived for ORB and LSD SLAM to allow a fair comparison with SVO. The first and second column report mean and standard devitation of the processing time. Since all algorithms use multi-threading, the third column reports the average CPU load when providing new images at a constant rate of 20 Hz.

detect large loop closures via image retrieval. Additionally, we provide results using both image streams of the stereo camera. Therefore, we apply the approach introduced in Sec. B.8 to estimate the motion of a multi-camera system.

To obtain a measure of accuracy of the different approaches, we align the final trajectory of keyframes with the ground-truth trajectory using the least-squares approach proposed in [Umeyama, 1991]. Since scale cannot be recovered using a single camera, we also rescale the estimated trajectory to best fit with the ground-truth trajectory. Subsequently, we compute the Euclidean distance between the estimated and ground-truth keyframe poses and compute the mean, median, and Root Mean Square Error (RMSE) in meters. We chose the absolute trajectory error measure instead of relative drift metrics [Sturm et al., 2012] because the final trajectory in ORB-SLAM consists only

	Thread	Intel i7 [ms]	Jetson TX1 [ms]
Sparse image alignment	1	0.66	2.54
Feature alignment	1	1.04	1.40
Optimize pose & landmarks	1	0.42	0.88
Extract features	2	1.64	5.48
Update depth filters	2	1.80	2.97

Table B.2 – Mean time consumption in milliseconds by individual components of SVO Mono on the EUROC Machine Hall 1 dataset. We report timing results on a laptop with Intel Core i7 (2.80 GHz) processor and on the NVIDIA Jetson TX1 ARM processor.

of a sparse set of keyframes, which makes drift measures on relatively short trajectories less expressive. The reported results are averaged over three runs.

The results show that using a stereo camera in general results in much higher accuracy. Apart from the additional visual measurements, the main reason for the improved results is that the stereo system does not drift in scale and inter camera triangulations allow to quickly initialize new 3D landmarks in case of on-spot rotations. Notice that SVO with bundle adjustment is twice as accurate as ORB and LSD SLAM.

However, the power of the less accurate configurations of SVO becomes obvious when analyzing the timing and processor usage of the approaches, which are reported in the right columns of Table B.1. In the table, we report the mean time to process a single frame in milliseconds and the standard deviation over all measurements. Since all algorithms make use of multi-threading and the time to process a single frame may therefore be misleading, we additionally report the CPU usage (continuously sampled during execution) when providing new images at a constant rate of 20 Hz to the algorithm. All measurements are averaged over 3 runs of the first EUROC dataset and computed on the same laptop computer (Intel Core i7-2760QM CPU). In Table E.1, we further report the average time consumption of individual components of SVO on the laptop computer and an NVIDIA TX1 ARM processor, which is popular in mobile robotics applications. The results show that the SVO approach is up to ten times faster than ORB-SLAM and LSD-SLAM and requires only a fourth of the CPU usage. The reason for this significant difference is that SVO does not extract features and descriptors in every frame, as in ORB-SLAM, but does so only for keyframes in the concurrent mapping thread. Additionally, ORB-SLAM—being a SLAM approach—spends most of the processing time in finding matches to the map (see Table I in [Mur-Artal et al., 2015b]), which in theory results in a pose-estimate without drift in an already mapped area. Contrarily, in the first three configurations of SVO, we estimate only the pose of the latest camera frame with respect to the local map. Compared to LSD-SLAM, SVO is faster because it operates on significantly less numbers of pixels, hence, also does not result in a semi-dense reconstruction of the environment. This, however, could be achieved in a parallel process as we have shown in [Pizzoli et al.,

	fr2_desk RMSE [cm]	fr2_xyz RMSE [cm]
SVO Mono (with edgelets)	9.7	1.1
SVO Mono + Bundle Adjustment	6.7	0.8
LSD-SLAM [Engel et al., 2014] ◯	4.5	1.5
ORB-SLAM [Mur-Artal et al., 2015a] ◯	**0.9**	0.3
PTAM [Klein and Murray, 2007]	× / ×	**0.2** / 24.3
Semi-Dense VO [Engel et al., 2013]	13.5	3.8
Direct RGB-D VO [Kerl et al., 2013] ★	1.8	1.2
Feature-based RGB-D SLAM [Endres et al., 2012] ★ ◯	9.5	2.6

Table B.3 – Results on the TUM RGB-D benchmark dataset [Sturm et al., 2012]. Results for [Engel et al., 2014, 2013, Kerl et al., 2013, Endres et al., 2012] were obtained from [Engel et al., 2014] and for PTAM we report two results that were published in [Mur-Artal et al., 2015a] and [Engel et al., 2013] respectively. Algorithms marked with ★ use a depth-sensor, and ◯ indicates loop-closure detection. The symbol × indicates that tracking the whole trajectory did not succeed.

2014, Faessler et al., 2015, Forster et al., 2015c]. We further remark that the bundle adjustment version has a significantly higher standard deviation in the timing as we update iSAM2 at every keyframe, which takes approximately 10 milliseconds longer than processing a regular frame. Using a motion prior further helps to improve the efficiency as the sparse-image-alignment optimization can be initialized closer to the solution, and therefore needs less iterations to converge.

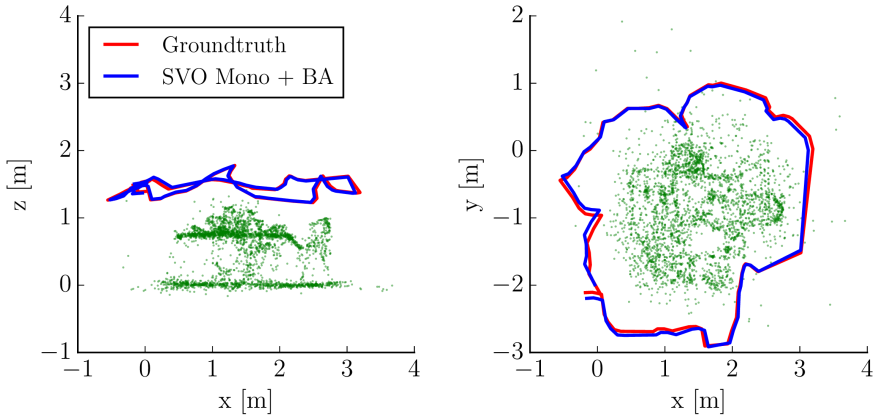
An edgelet provides only a one-dimensional constraint in the image domain, while a corner provides a two-dimensional constraint. Therefore, whenever sufficient corners can be detected, the SVO algorithm prioritizes the corners. Since the environment in the EUROC dataset is well textured and provides many corners, the use of edgelets does not improve the accuracy. However, the edgelets bring a benefit in terms of robustness when the texture is such that no corners are present.

TUM Datasets

A common dataset to evaluate visual odometry algorithms is the TUM Munich RGB-D benchmark [Sturm et al., 2012]. The dataset was recorded with a Microsoft Kinect RGB-D camera, which provides images of worse quality (e.g. rolling shutter, motion blur) than the VI-Sensor EUROC dataset. Fig. B.13 shows excerpts from the "`fr2_desk`" and "`fr2_xyz`" datasets which have a trajectory length of 18.8 m and 7 m respectively. Groundtruth is provided by a motion capture system. Table B.3 shows the results of the proposed algorithm (averaged over three runs) and comparisons against related works. The resulting trajectory and the recovered landmarks are shown in Fig. B.14. The results from related works were obtained from the evaluation in [Mur-Artal et al.,

(a) fr2 desk (b) fr2 xyz

Figure B.13 – Impressions from the TUM RGB-D benchmark dataset Sturm et al. [2012] with tracked corners in green and edgelets in magenta.



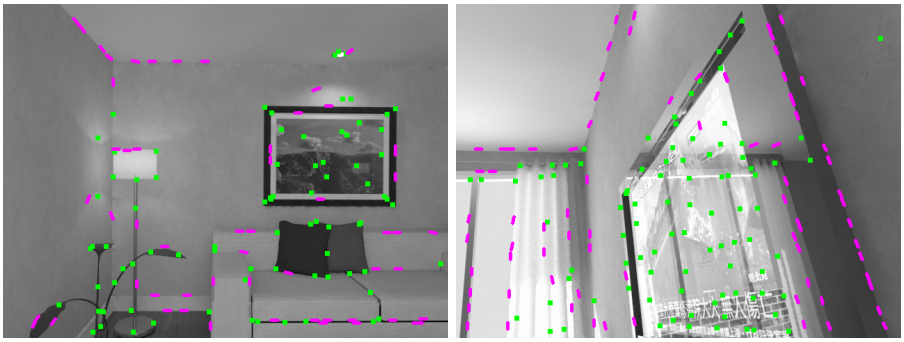
(a) fr2_desk (side) (b) fr2_desk (top)

Figure B.14 – Estimated trajectory and pointcloud of the TUM "fr2_desk" dataset.

2015a] and [Engel et al., 2014]. We argue that the better performance of ORB-SLAM and LSD-SLAM is due to the capability to detect loop-closures.

ICL-NUIM Datasets

The ICL-NUIM dataset [Handa et al., 2014] is a synthetic dataset that aims to benchmark RGB-D, visual odometry and SLAM algorithms. The dataset consists of four trajectories of length 6.4 m, 1.9 m, 7.3 m, and 11.1 m. The synthesized images are corrupted by noise to simulate real camera images. Ground-truth and calibration are provided by the dataset. Most reported results on this dataset use the synthetsized measurements from the depth sensor together with the rendered images. Indeed, the datsets are very challenging for purely vision-based odometry due to difficult texture and frequent on-spot rotations as can be seen in the excerpts from the dataset in Fig. B.15.



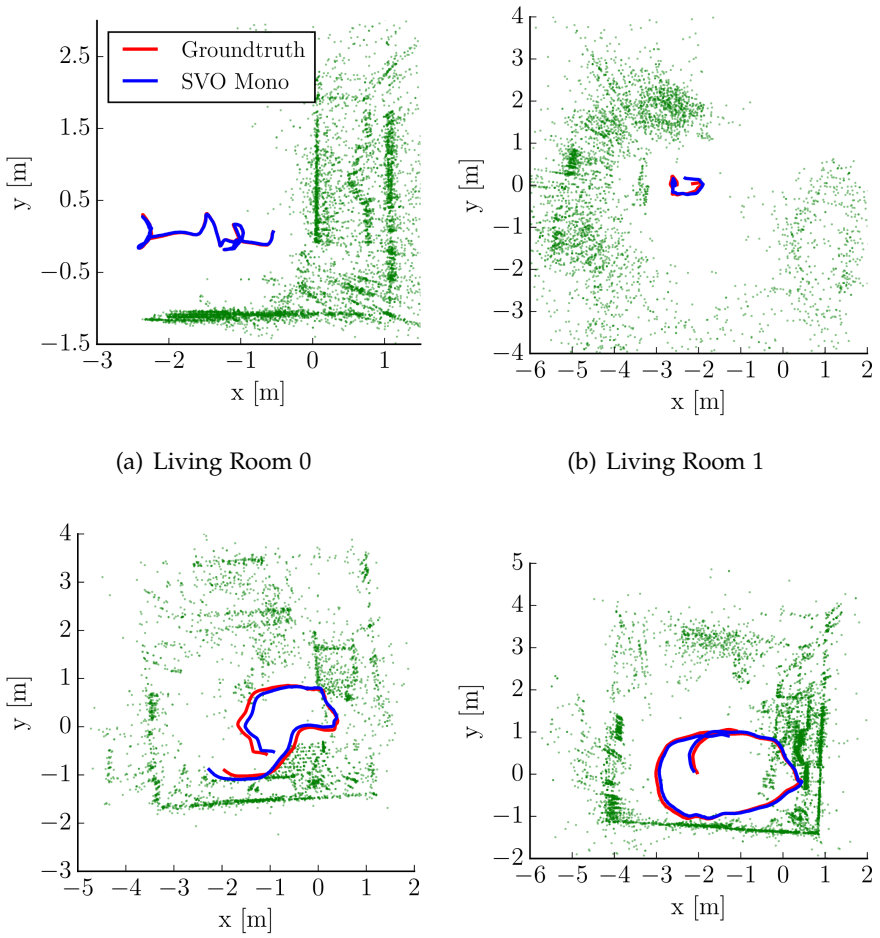
(a) (b)

Figure B.15 – Impressions from the synthetic ICL-NUIM dataset Handa et al. [2014] with tracked corners marked in green and edgelets in magenta.

	lr_kt0 RMSE	lr_kt1 RMSE	lr_kt2 RMSE	lr_kt3 RMSE
SVO Mono (with edgelets)	**0.02**	0.07	0.1	**0.07**
LSD SLAM	×	×	×	×
ORB SLAM [Mur-Artal et al., 2015a] ◯	×	×	0.03	0.12
DVO [Kerl et al., 2013] ★	0.29	0.12	0.47	0.54
FOVIS [Huang et al., 2011a] ★	2.05	1.87	1.49	1.47
ICP [Newcombe et al., 2011a] ★	0.07	**0.005**	**0.01**	0.36
ICP+RGB-D [Whelan et al., 2013] ★	0.39	0.021	0.12	0.86

Table B.4 – Results on the ICL-NUIM Dataset [Handa et al., 2014]. Algorithms marked with ★ use a depth-sensor, and ◯ indicates loop-closure detection. The symbol × indicates that tracking the whole trajectory did not succeed.

Table B.4 reports the results of the proposed algorithm (averaged over three runs) and the results from other algorithms on the "living room" sequence. Similar to the previous datasets, we report the root mean square error after rotation, translation and scale alignment with the ground-truth trajectory. Fig. B.16 shows the reconstructed maps and recovered trajectories. The maps are very noisy due to the fine grained texture of the scene. We also run ORB-SLAM and LSD-SLAM on the dataset. ORB-SLAM fails to initialize on the second sequence and we did not manage to obtain any results from LSD-SLAM on all datasets. The reason for the failures is lack of texture for initialization and frequent on-spot rotations. For SVO, this also required us to set a particularly low FAST corner detection threshold on this dataset (to 5 instead of 20). A lower threshold results in detection of many low-quality features. However, features are only used in SVO once their corresponding scene depth is sucessfully estimated by means of the robust depth filter described in Sec. B.6. Hence, the process of depth estimation helps to identify the stable features with low score that can be reliably used for motion estimation.



(a) Living Room 0

(b) Living Room 1

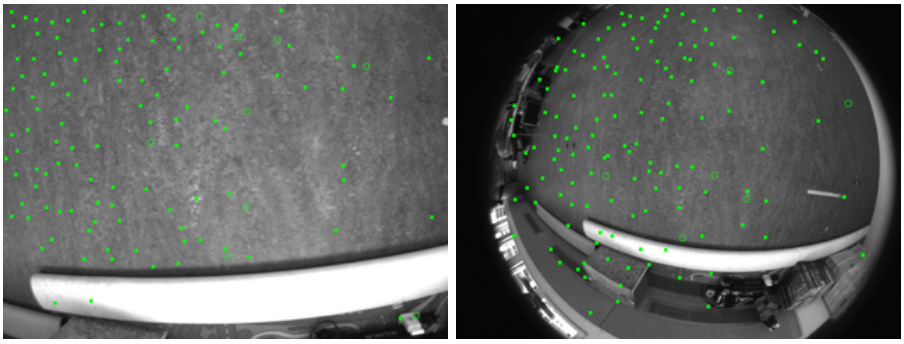
(c) Living Room 2

(d) Living Room 3

Figure B.16 – Results on the ICL-NUIM Handa et al. [2014] noisy synthetic living room dataset.

In this dataset, we were not able to refine the results of SVO with bundle adjustment. The reason is that the iSAM2 backend is based on Gauss Newton which is very sensitive to underconstrained variables that render the linearized problem indeterminant. The frequent on-spot rotations and very low parallax angle triangulations result in many underconstrained variables. Using an optimizer that is based on Levenberg Marquardt or adding additional inertial measurements [Forster et al., 2015a] would help in such cases.

For comparison, Table B.4 also shows the results reported in [Handa et al., 2014] of algorithms that also use the depth measurements.



(a) Perspective. (b) Fisheye.

Figure B.17 – SVO tracking with (a) perspective and (b) fisheye camera lens.

Circle Dataset

In the last experiment, we want to demonstrate the usefulness of wide field of view lenses for VO. We recorded the dataset with a micro aerial vehicle that we flew in a motion capture room and commanded it to fly a perfect circle with downfacing camera. Subsequently, we flew the exact same trajectory again with a wide fisheye camera. Excerpts from the dataset are shown in Fig. B.17. We run SVO (without bundle adjustment) on both datasets and show the resulting trajectories in Fig. B.18. To run SVO on the fisheye images, we use the modifications described in Sec. B.7. While the recovered trajectory from the perspective camera slowly drifts over time, the result on the fisheye camera perfectly overlaps with the groundtruth trajectory. We also run ORB-SLAM and LSD-SLAM on the trajectory with the perspective images. The result of ORB-SLAM is as close to the ground-truth trajectory as the SVO fisheye result. However, if we deactivate loop-closure detection (shown result) the trajectory drifts more than SVO. We were not able to run LSD-SLAM and ORB-SLAM on the fisheye images as the open source implementations do not support very large FoV cameras. Due to the difficult high-frequency texture of the floor, we were not able to initialize LSD-SLAM on this dataset. A more in-depth evaluation of the benefit of large FoV cameras for SVO is provided in Zhang et al. [2016].

Discussion

In this section we discuss the proposed SVO algorithm in terms of efficiency, accuracy, and robustness.

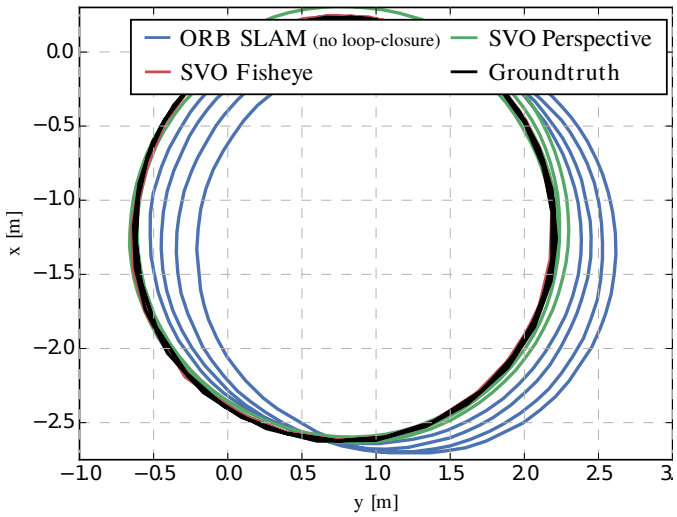


Figure B.18 – Comparison of perspective and fisheye lense on the same circular trajectory that was recorded with a micro aerial vehicle in a motion capture room. The ORB-SLAM result was obtained with the perspective camera images and loop-closure was deactivated for a fair comparison with SVO. ORB-SLAM with a perspective camera and with loop-closure activated performs as good as SVO with a fisheye camera.

Efficiency

Feature-based algorithms incur a constant cost of feature and descriptor extraction per frame. For example, ORB-SLAM requires 11 milliseconds per frame for ORB feature extraction only [Mur-Artal et al., 2015a]. This constant cost per frame is a bottleneck for feature-based VO algorithms. On the contrary, SVO does not have this constant cost per frame and benefits greatly from the use of high frame-rate cameras. SVO extracts features only for selected keyframes in a parallel thread, thus, decoupled from hard real-time constraints. The proposed tracking algorithm, on the other hand, benefits from high frame-rate cameras: the sparse image alignment step is automatically initialized closer to the solution and, thus, converges faster. Therefore, increasing the camera frame-rate actually reduces the computational cost per frame in SVO. The same principle applies to LSD-SLAM. However, LSD-SLAM tracks significantly more pixels than SVO and is, therefore, up to an order of magnitude slower. To summarize, on a laptop computer with an Intel i7 2.8 GHz CPU processor ORB-SLAM and LSD-SLAM require approximately 30 and 23 milliseconds respectively per frame while SVO requires only 2.5 milliseconds (see Table B.1).

Accuracy

SVO computes feature correspondence with sub-pixel accuracy using direct feature alignment. Subsequently, we optimize both structure and motion to minimize the

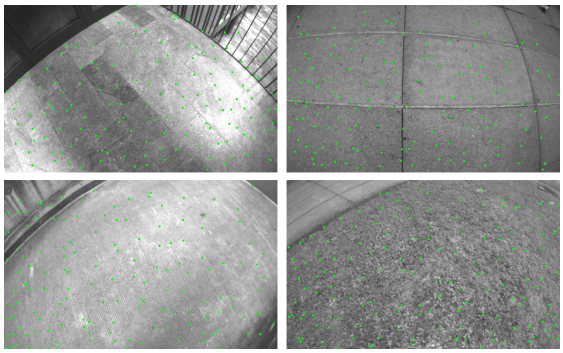


Figure B.19 – Successful tracking in scenes of high-frequency texture.

reprojection errors (see Sec. B.5.2). We use SVO in two settings: if highest accuracy is not necessary, such as for motion estimation of micro aerial vehicles [Faessler et al., 2015], we only perform the refinement step (Sec. B.5.2) for the latest camera pose, which results in the highest frame-rates (*i.e.*, 2.5 ms). If highest accuracy is required, we use iSAM2 [Kaess et al., 2012] to jointly optimize structure and motion of the whole trajectory. iSAM2 is an incremental smoothing algorithm, which leverages the expressiveness of factor graphs [Dellaert and Kaess, 2006] to maintain sparsity and to identify and update only the typically small subset of variables affected by a new measurement. In an odometry setting, this allows iSAM2 to achieve the same accuracy as batch estimation of the whole trajectory, while preserving real-time capability. Bundle adjustment with iSAM2 is consistent [Forster et al., 2015a], which means that the estimated covariance of the estimate matches the estimation errors (*e.g.*, are not over-confident). Consistency is a prerequisite for optimal fusion with additional sensors [Bar-Shalom et al., 2001]. In [Forster et al., 2015a], we therefore show how SVO can be fused with inertial measurements to achieve a drift that is approximately 0.1% of the traveled distance. LSD-SLAM, on the other hand, only optimizes a graph of poses and leaves the structure fixed once computed (up to a scale). The optimization does not capture correlations between the semi-dense depth estimates and the camera pose estimates. This separation of depth estimation and pose optimization is only optimal if each step yields the optimal solution.

Robustness

SVO is most robust when a high frame-rate camera is used (*e.g.*, between 40 and 80 frames per second). This increases the resilience to fast motions as it is demonstrated in the video attachment. A fast camera, together with the proposed robust depth estimation, allows us to track the camera in environments with repetitive and high frequency texture (*e.g.*, grass or asphalt as shown in Fig. B.19). The advantage of the proposed probabilistic depth estimation method over the standard approach of triangulating points from two views only is that we observe far fewer outliers as every depth filter undergoes many measurements until convergence. Furthermore, erroneous

measurements are explicitly modeled, which allows the depth to converge in highly self-similar environments.

A further advantage of SVO is that the algorithm starts directly with an optimization. Data association in sparse image alignment is directly given by geometry of the problem and therefore, no RANSAC [Fischler and Bolles, 1981] is required as it is typical in feature-based approaches. Starting directly with an optimization also simplifies greatly the use of multi-camera systems, which greatly improves resilience to on-spot rotations as the field of view of the system is enlarged and depth can be triangulated from inter-camera-rig measurements.

Finally, the use of gradient edge features (*i.e.*, edgelets) increases the robustness in areas where only few corner features are found. Our simulation experiments have shown that the proposed sparse image alignment approach achieves comparable performance as semi-dense and dense alignment in terms of robustness of frame-to-frame motion estimation.

Conclusion

In this paper, we proposed the semi-direct VO pipeline "SVO" that is significantly faster than the current state-of-the-art VO algorithms while achieving highly competitive accuracy. The gain in speed is due to the fact that features are only extracted for selected keyframes in a parallel thread and feature matches are established very fast and robustly with the novel sparse image alignment algorithm. Sparse image alignment tracks a set of features jointly under epipolar constrains and can be used instead of KLT-tracking [Lucas and Kanade, 1981] when the scene depth at the feature positions is known. We further propose to estimate the scene depth using a robust filter that explicitly models outlier measurements. Robust depth estimation and direct tracking allows us to track very weak corner features and edgelets. A further benefit of SVO is that it directly starts with an optimization, which allows us to easily integrate measurements from multiple cameras as well as motion priors. The formulation further allows using large FoV cameras with fisheye and catadioptric lenses. The SVO algorithm has further proven successful in real-world applications such as vision-based flight of quadrotors [Faessler et al., 2015] or 3D scanning applications with smartphones. **Acknowledgments** The authors gratefully acknowledge Henri Rebecq for creating the "Urban Canyon" datasets that can be accessed here: http://rpg.ifi.uzh.ch/fov.html

Appendix

In this section, we derive the analytic solution to the multi-camera sparse-image-alignment problem with motion prior.

Given a rig of M calibrated cameras $c \in \mathrm{C}$ with known extrinsic calibration $\mathtt{T}_{\mathrm{CB}}$, the goal is to estimate the incremental body motion $\mathtt{T}_{\mathrm{BB}-1}$ by minimizing the intensity residual $\mathbf{r}_{\mathtt{I}_i^{\mathrm{C}}}$ of corresponding pixels in subsequent images. Corresponding pixels are found by means of projecting a known point on the scene surface $\boldsymbol{\rho}_i \doteq {}_{\mathrm{B}-1}\boldsymbol{\rho}_i$ (prefix $\mathrm{B}-1$ denotes that the point is expressed in the previous frame of reference) into images of camera C that were recorded at poses k and $k-1$, which are denoted $\mathtt{I}_k^{\mathrm{C}}$ and $\mathtt{I}_{k-1}^{\mathrm{C}}$ respectively. To improve the convergence properties of the optimization (see Sec. B.11.1), we accumulate the intensity residual errors in small patches $\mathcal{P}$ centered at the pixels where the 3d points project. Therefore, we use the iterator variable $\Delta\mathbf{u}$ to sum the intensities over a small patch $\mathcal{P}$. We further assume that a prior of the incremental body motion $\tilde{\mathtt{T}}_{kk-1} \doteq (\tilde{\mathtt{R}}, \tilde{\mathbf{p}})$ is given. The goal is to find the incremental camera rotation and translation $\mathtt{T}_{kk-1} \doteq (\mathtt{R}, \mathbf{p})$ that minimizes the sum of squared errors:

$$
\begin{aligned}
(\mathtt{R}^\star, \mathbf{p}^\star) &= \arg\min_{(\mathtt{R},\mathbf{p})} C(\mathtt{R}, \mathbf{p}), \quad \text{with} \\
C(\mathtt{R}, \mathbf{p}) &= \sum_{\mathrm{C}\in\mathrm{C}} \sum_{i=1}^{N} \sum_{\Delta\mathbf{u}\in\mathcal{P}} \frac{1}{2} \|\mathbf{r}_{\mathtt{I}^{\mathrm{C}}_{i,\Delta\mathbf{u}}}\|^2_{\Sigma_{\mathtt{I}}} + \frac{1}{2}\|\mathbf{r}_{\mathtt{R}}\|^2_{\Sigma_{\mathtt{R}}} + \frac{1}{2}\|\mathbf{r}_{\mathbf{p}}\|^2_{\Sigma_{\mathbf{p}}},
\end{aligned}
\tag{B.13}
$$

where N is the number of visible 3D points. We have furder defined the image intensity and prior residuals as:

$$
\begin{aligned}
\mathbf{r}_{\mathtt{I}^{\mathrm{C}}_{i,\Delta\mathbf{u}}} &\doteq \mathtt{I}^{\mathrm{C}}_k\Big(\pi(\mathtt{T}_{\mathrm{CB}}(\mathtt{R}\boldsymbol{\rho}_i + \mathbf{p})) + \Delta\mathbf{u}\Big) - \mathtt{I}^{\mathrm{C}}_{k-1}\Big(\pi(\mathtt{T}_{\mathrm{CB}}\,\boldsymbol{\rho}_i) + \Delta\mathbf{u}\Big) \\
\mathbf{r}_{\mathtt{R}} &\doteq \log(\tilde{\mathtt{R}}^{\mathsf{T}}\mathtt{R})^\vee \\
\mathbf{r}_{\mathbf{p}} &\doteq \mathbf{p} - \tilde{\mathbf{p}}
\end{aligned}
\tag{B.14}
$$

For readability, we write the cost function in matrix form

$$
C(\mathtt{R}, \mathbf{p}) = \mathbf{r}(\mathtt{R}, \mathbf{p})^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} \mathbf{r}(\mathtt{R}, \mathbf{p}),
\tag{B.15}
$$

where $\boldsymbol{\Sigma}$ is a block-diagonal matrix composed of the measurement covariances. Since the residuals are non-linear in $(\mathtt{R}, \mathbf{p})$, we solve the optimization problem in an iterative Gauss-Newton procedure [Barfoot, 2015]. Therefore, we substitute the following perturbations in the cost function:

$$
\mathtt{R} \;\leftarrow\; \mathtt{R}\,\exp(\delta\boldsymbol{\phi}^\wedge), \qquad \mathbf{p} \;\leftarrow\; \mathbf{p} + \mathtt{R}\delta\mathbf{p},
\tag{B.16}
$$

where the hat operator $(.)^\wedge$ forms a 3×3 skew-symmetric matrix from a vector in $\mathbb{R}^3$.

As it is common practice for optimizations involving rotations [Forster et al., 2015a, Barfoot, 2015], we use the exponential map $\exp(\cdot)$ to perturb the rotation in the tangent space of $\mathrm{SO}(3)$ which avoids singularities and provides a minimal parametrization of the rotation increment. The *exponential map* (at the identity) $\exp : \mathfrak{so}(3) \to \mathrm{SO}(3)$ associates a 3×3 skew-symmetric matrix to a rotation and coincides with the standard

matrix exponential (Rodrigues' formula):

$$\exp(\boldsymbol{\phi}^\wedge) = \mathbf{I} + \frac{\sin(\|\boldsymbol{\phi}\|)}{\|\boldsymbol{\phi}\|}\boldsymbol{\phi}^\wedge + \frac{1-\cos(\|\boldsymbol{\phi}\|)}{\|\boldsymbol{\phi}\|^2}\left(\boldsymbol{\phi}^\wedge\right)^2. \tag{B.17}$$

The inverse relation is the *logarithm map* (at the identity), which associates $\mathtt{R} \in \mathrm{SO}(3)$ to a skew symmetric matrix:

$$\log(\mathtt{R}) = \frac{\varphi \cdot (\mathtt{R} - \mathtt{R}^\mathsf{T})}{2\sin(\varphi)} \text{ with } \varphi = \cos^{-1}\left(\frac{\mathrm{tr}\,(\mathtt{R}) - 1}{2}\right). \tag{B.18}$$

Note that $\log(\mathtt{R})^\vee = \mathbf{a}\varphi$, where $\mathbf{a}$ and φ are the rotation axis and the rotation angle of $\mathtt{R}$, respectively.

Substituting the perturbations makes the residual errors a function defined on a vector space. This allows us to linearize the quadratic cost at the current estimate, form the *normal equations*, and solve them for the optimal perturbations:

$$\mathbf{J}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{J}\,[\delta\boldsymbol{\phi}^\mathsf{T}\ \delta\mathbf{p}^\mathsf{T}]^\mathsf{T} = -\mathbf{J}^\mathsf{T}\boldsymbol{\Sigma}^{-1}\mathbf{r}(\mathtt{R},\mathbf{p}), \tag{B.19}$$

where we introduced the variable $\mathbf{J}$, which stacks all Jacobian matrices from the linearization. The solution is subsequently used to update our estimate in $(\mathtt{R},\mathbf{p})$ according to (B.16). This procedure is repeated until the norm of the update vectors is sufficiently small, which indicates convergence.

In the following, we show how to linearize the residuals to obtain the Jacobians. Therefore, we substitute the perturbations in the residuals and expand:

$$\begin{aligned}&\mathbf{r}_\mathtt{R}(\mathtt{R}\,\exp(\delta\boldsymbol{\phi}^\wedge)) \\ &\quad= \log(\tilde{\mathtt{R}}^\mathsf{T}\mathtt{R}\,\exp(\delta\boldsymbol{\phi}^\wedge))^\vee \overset{(a)}{\simeq} \mathbf{r}_\mathtt{R}(\mathtt{R}) + \mathtt{J}_r^{-1}(\log(\tilde{\mathtt{R}}^\mathsf{T}\mathtt{R})^\vee)\delta\boldsymbol{\phi}\end{aligned} \tag{B.20}$$

$$\begin{aligned}&\mathbf{r}_\mathbf{p}(\mathbf{p} + \mathtt{R}\delta\mathbf{p}) \\ &\quad= (\mathbf{p} + \mathtt{R}\delta\mathbf{p}) - \tilde{\mathbf{p}} = \mathbf{r}_\mathbf{p}(\mathbf{p}) + \mathtt{R}\delta\mathbf{p}\end{aligned} \tag{B.21}$$

$$
\begin{aligned}
&\mathbf{r}_{\mathtt{I}_i^{\mathtt{C}}}(\mathtt{R}\ \exp(\delta\boldsymbol{\phi}^\wedge)) \\
&= \mathtt{I}_k^{\mathtt{C}}\Big(\pi(\mathtt{T}_{\mathtt{CB}}(\mathtt{R}\ \exp(\delta\boldsymbol{\phi}^\wedge)\boldsymbol{\rho}_i + \mathbf{p}))\Big) - \mathtt{I}_{k-1}^{\mathtt{C}}\Big(\pi(\mathtt{T}_{\mathtt{CB}}\,\boldsymbol{\rho}_i)\Big) \\
&\overset{(b)}{\simeq} \mathtt{I}_k^{\mathtt{C}}\Big(\pi(\mathtt{T}_{\mathtt{CB}}(\mathtt{R}\boldsymbol{\rho}_i + \mathbf{p}))\Big) - \mathtt{I}_{k-1}^{\mathtt{C}}\Big(\pi(\mathtt{T}_{\mathtt{CB}}\exp(\delta\boldsymbol{\phi}^\wedge)^{-1}\boldsymbol{\rho}_i)\Big) \\
&\overset{(c)}{\simeq} \mathtt{I}_k^{\mathtt{C}}\Big(\pi(\mathtt{T}_{\mathtt{CB}}(\mathtt{R}\boldsymbol{\rho}_i + \mathbf{p}))\Big) - \mathtt{I}_{k-1}^{\mathtt{C}}\Big(\pi(\mathtt{T}_{\mathtt{CB}}(\mathbf{I} - \delta\boldsymbol{\phi}^\wedge)\boldsymbol{\rho}_i)\Big) \\
&\overset{(d)}{=} \mathtt{I}_k^{\mathtt{C}}\Big(\pi(\mathtt{T}_{\mathtt{CB}}(\mathtt{R}\boldsymbol{\rho}_i + \mathbf{p}))\Big) - \mathtt{I}_{k-1}^{\mathtt{C}}\Big(\pi(\mathtt{T}_{\mathtt{CB}}\boldsymbol{\rho}_i + \mathtt{T}_{\mathtt{CB}}\boldsymbol{\rho}_i^\wedge\delta\boldsymbol{\phi})\Big) \\
&\overset{(e)}{\simeq} \mathbf{r}_{\mathtt{I}_i^{\mathtt{C}}}(\mathtt{R}) - \frac{\partial \mathtt{I}_{k-1}^{\mathtt{C}}(\mathbf{u})}{\partial \mathbf{u}}\bigg|_{\mathbf{u}=\pi(_c\boldsymbol{\rho}_i)} \frac{\partial \pi(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}}\bigg|_{\boldsymbol{\rho}=_c\boldsymbol{\rho}_i} \mathtt{R}_{\mathtt{CB}}\boldsymbol{\rho}_i^\wedge\delta\boldsymbol{\phi}
\end{aligned}
\tag{B.22}
$$

$$
\begin{aligned}
&\mathbf{r}_{\mathtt{I}_i^{\mathtt{C}}}(\mathbf{p} + \mathtt{R}\delta\mathbf{p}) \\
&= \mathtt{I}_k^{\mathtt{C}}\Big(\pi(\mathtt{T}_{\mathtt{CB}}(\mathtt{R}\boldsymbol{\rho}_i + \mathbf{p} + \mathtt{R}\delta\mathbf{p}))\Big) - \mathtt{I}_{k-1}^{\mathtt{C}}\Big(\pi(\mathtt{T}_{\mathtt{CB}}\,\boldsymbol{\rho}_i)\Big) \\
&\overset{(b)}{\simeq} \mathtt{I}_k^{\mathtt{C}}\Big(\pi(\mathtt{T}_{\mathtt{CB}}(\mathtt{R}\boldsymbol{\rho}_i + \mathbf{p}))\Big) - \mathtt{I}_{k-1}^{\mathtt{C}}\Big(\pi(\mathtt{T}_{\mathtt{CB}}(\boldsymbol{\rho}_i - \delta\mathbf{p}))\Big) \\
&\overset{(e)}{\simeq} \mathbf{r}_{\mathtt{I}_i^{\mathtt{C}}}(\mathtt{R}) + \frac{\partial \mathtt{I}_{k-1}^{\mathtt{C}}(\mathbf{u})}{\partial \mathbf{u}}\bigg|_{\mathbf{u}=\pi(_c\boldsymbol{\rho}_i)} \frac{\partial \pi(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}}\bigg|_{\boldsymbol{\rho}=_c\boldsymbol{\rho}_i} \mathtt{R}_{\mathtt{CB}}\delta\mathbf{p}
\end{aligned}
\tag{B.23}
$$

In step (a), we have used a first-order expansion of the matrix logarithm:

$$
\log\Big(\exp(\boldsymbol{\phi}^\wedge)\ \exp(\delta\boldsymbol{\phi}^\wedge)\Big)^\vee \approx \boldsymbol{\phi} + \mathtt{J}_r^{-1}(\boldsymbol{\phi})\delta\boldsymbol{\phi}, \tag{B.24}
$$

which holds for small values of $\delta\boldsymbol{\phi}$. The term $\mathtt{J}_r^{-1}$ is the inverse of the *right Jacobian of* SO(3) [Barfoot, 2015, Chirikjian, 2012]:

$$
\mathtt{J}_r^{-1}(\boldsymbol{\phi}) = \mathbf{I} + \frac{1}{2}\boldsymbol{\phi}^\wedge + \left(\frac{1}{\|\boldsymbol{\phi}\|^2} + \frac{1+\cos(\|\boldsymbol{\phi}\|)}{2\|\boldsymbol{\phi}\|\sin(\|\boldsymbol{\phi}\|)}\right)(\boldsymbol{\phi}^\wedge)^2.
$$

In step (b), we invert the perturbation and apply it to the reference frame. This trick stems from the *inverse compositional* [Baker and Matthews, 2004] formulation, which allows us to keep the term containing the perturbation constant such that the Jacobian of the intensity residual remains unchanged over all iterations, greatly improving compuational efficiency. In (c), we first used that $\exp(\delta\boldsymbol{\phi}^\wedge)^{-1} = \exp(-\delta\boldsymbol{\phi}^\wedge)$ and subsequently used the first-order approximation of the exponential map:

$$
\exp(\delta\boldsymbol{\phi}) \simeq \mathbf{I} + \delta\boldsymbol{\phi}^\wedge. \tag{B.25}
$$

For step (d), we used a property of skew symmetric matrices

$$
\delta\boldsymbol{\phi}^\wedge\boldsymbol{\rho} = -\boldsymbol{\rho}^\wedge\delta\boldsymbol{\phi}. \tag{B.26}
$$

Finally, in step (e), we perform a Taylor expansion around the perturbation. The term

$\frac{\partial \mathtt{I}^{\mathtt{C}}_{k-1}(\mathbf{u})}{\partial \mathbf{u}}$ denotes the image derivative at pixel $\mathbf{u}$ and $\frac{\partial \pi(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}}$ is the derivative of the camera projection function, which for standard pinhole projection with focal length (f_x, f_y) and camera center (c_x, c_y) takes the form

$$\frac{\partial \pi(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}} = \begin{bmatrix} f_x \frac{x}{z} & 0 & -\frac{c_x}{z^2} \\ 0 & f_y \frac{y}{z} & -\frac{c_y}{z^2} \end{bmatrix} \quad \text{with } \boldsymbol{\rho} = [x, y, z]^\mathsf{T}. \tag{B.27}$$

To summarize, the Jacobians of the residuals are:

$$\begin{aligned}
\frac{\partial \mathbf{r}_{\mathtt{R}}}{\partial \delta \boldsymbol{\phi}} &= \mathtt{J}_r^{-1}(\mathrm{Log}(\tilde{\mathtt{R}}^\mathsf{T}\mathtt{R})) \\
\frac{\partial \mathbf{r}_{\mathbf{p}}}{\partial \delta \mathbf{p}} &= \mathtt{R} \\
\frac{\partial \mathbf{r}_{\mathtt{I}^{\mathtt{C}}_i}}{\partial \delta \boldsymbol{\phi}} &= -\frac{\partial \mathtt{I}^{\mathtt{C}}_{k-1}(\mathbf{u})}{\partial \mathbf{u}}\bigg|_{\mathbf{u}=\pi({}_c\boldsymbol{\rho}_i)} \frac{\partial \pi(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}}\bigg|_{\boldsymbol{\rho}={}_c\boldsymbol{\rho}_i} \mathtt{R}_{\mathrm{CB}} \boldsymbol{\rho}_i^{\wedge} \\
\frac{\partial \mathbf{r}_{\mathtt{I}^{\mathtt{C}}_i}}{\partial \delta \mathbf{p}} &= \frac{\partial \mathtt{I}^{\mathtt{C}}_{k-1}(\mathbf{u})}{\partial \mathbf{u}}\bigg|_{\mathbf{u}=\pi({}_c\boldsymbol{\rho}_i)} \frac{\partial \pi(\boldsymbol{\rho})}{\partial \boldsymbol{\rho}}\bigg|_{\boldsymbol{\rho}={}_c\boldsymbol{\rho}_i} \mathtt{R}_{\mathrm{CB}}
\end{aligned} \tag{B.28}$$

C Visual-Inertial Estimation

C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. On-Manifold Preintegration for Real-Time Visual-Inertial Odometry. *IEEE Transactions on Robotics (TRO)* URL http://rpg.ifi.uzh.ch/docs/TRO16_forster.pdf.

A shorter version of this article was previously published as:

C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. In *Robotics: Science and Systems (RSS)*, 2015. URL http://dx.doi.org/10.15607/RSS.2015.XI.006.

On-Manifold Preintegration for Real-Time Visual-Inertial Odometry

Christian Forster, Luca Carlone, Frank Dellaert, Davide Scaramuzza

Abstract — Current approaches for visual-inertial odometry (VIO) are able to attain highly accurate state estimation via nonlinear optimization. However, real-time optimization quickly becomes infeasible as the trajectory grows over time; this problem is further emphasized by the fact that inertial measurements come at high rate, hence leading to fast growth of the number of variables in the optimization. In this paper, we address this issue by preintegrating inertial measurements between selected keyframes into single relative motion constraints. Our first contribution is a *preintegration theory* that properly addresses the manifold structure of the rotation group. We formally discuss the generative measurement model as well as the nature of the rotation noise and derive the expression for the *maximum a posteriori* state estimator. Our theoretical development enables the computation of all necessary Jacobians for the optimization and a-posteriori bias correction in analytic form. The second contribution is to show that the preintegrated IMU model can be seamlessly integrated into a visual-inertial pipeline under the unifying framework of factor graphs. This enables the application of incremental-smoothing algorithms and the use of a *structureless* model for visual measurements, which avoids optimizing over the 3D points, further accelerating the computation. We perform an extensive evaluation of our monocular VINpipeline on real and simulated datasets. The results confirm that our modelling effort leads to accurate state estimation in real-time, outperforming state-of-the-art approaches.

Introduction

The use of cameras and inertial sensors for three-dimensional structure and motion estimation has received considerable attention from the robotics community. Both sensor types are cheap, ubiquitous, and complementary. A single moving camera is an exteroceptive sensor that allows us to measure appearance and geometry of a three-dimensional scene, up to an unknown metric scale; an inertial measurement unit (IMU) is a proprioceptive sensor that renders metric scale of monocular vision and gravity observable [Martinelli, 2012] and provides robust and accurate inter-frame motion estimates. Applications of VINrange from autonomous navigation in GPS-denied environments, to 3D reconstruction, and augmented reality.

The existing literature on VIO imposes a trade-off between accuracy and computational efficiency (a detailed review is given in Section C.2). On the one hand, filtering approaches enable fast inference, but their accuracy is deteriorated by the accumulation of linearization errors. On the other hand, full smoothing approaches, based on nonlinear optimization, are accurate, but computationally demanding. Fixed-lag smoothing offers a compromise between accuracy for efficiency; however, it is not clear how to set the length of the estimation window so to guarantee a given level of performance.

In this work we show that it is possible to overcome this trade-off. We design a VINsystem that enables fast incremental smoothing and computes the optimal *maximum a posteriori* (MAP) estimate in real time. An overview of our approach is given in Section C.4.

The first step towards this goal is the development of a novel preintegration theory. The use of *preintegrated IMU measurements* was first proposed in [Lupton and Sukkarieh, 2012] and consists of combining many inertial measurements between two keyframes into a single relative motion constraint. We build upon this work and present a preintegration theory that properly addresses the manifold structure of the rotation group $\mathrm{SO}(3)$. Our preintegration theory is presented in Sections C.5-C.6. Compared with [Lupton and Sukkarieh, 2012], our theory offers a more formal treatment of the rotation noise, and avoids singularities in the representation of rotations. Furthermore, we are able to derive all necessary Jacobians in analytic form: specifically, we report the analytic Jacobians of the residuals, the noise propagation, and the a-posteriori bias correction in the appendix of this paper.

Our second contribution is to frame the IMU preintegration theory into a factor graph model. This enables the application of incremental smoothing algorithms, as iSAM2 [Kaess et al., 2012], which avoid the accumulation of linearization errors and offer an elegant way to trade-off accuracy with efficiency. Inspired by [Carlone et al., 2014, Mourikis and Roumeliotis, 2007], we also adopt a *structureless* model for visual

measurements, which allows eliminating a large number of variables (*i.e.*, all 3D points) during incremental smoothing, further accelerating the computation (Section C.7). In contrast to [Mourikis and Roumeliotis, 2007], we use the structureless model in an incremental smoothing framework. This has two main advantages: we do not need to delay the processing of visual measurements, and we can relinearize the visual measurements multiple times.

In order to demonstrate the effectiveness of our model, we integrated the proposed IMU preintegration in a state-of-the-art VINpipeline and tested it on real and simulated datasets (Sections C.8). Our theoretical development leads to tangible practical advantages: an implementation of the approach proposed in this paper performs full-smoothing at a rate of 100 Hz and achieves superior accuracy with respect to competitive state-of-the-art filtering and optimization approaches.

Besides the technical contribution, the paper also provides a tutorial contribution for practitioners. In Section C.3 and across the paper, we provide a short but concise summary of uncertainty representation on manifolds and exemplary derivations for uncertainty propagation and Jacobian computation. The complete derivation of all equations and Jacobians – necessary to implement our model – are given in the appendix.

This paper is an extension of our previous work [Forster et al., 2015b] with additional experiments, an in-depth discussion of related work, and comprehensive technical derivations. The results of the new experiments highlight the accuracy of bias estimation, demonstrate the consistency of our approach, and provide comparisons against full batch estimation. We release our implementation of the preintegrated IMU and structureless vision factors in the GTSAM 4.0 optimization toolbox [Dellaert, 2012].

Related Work

Related work on visual-inertial odometry can be sectioned along three main dimensions. The first dimension is the number of camera-poses involved in the estimation. While *full smoothers* (or *batch nonlinear least-squares* algorithms) estimate the complete history of poses, *fixed-lag smoothers* (or *sliding window estimators*) consider a window of the latest poses, and *filtering* approaches only estimate the latest state. Both fixed-lag smoothers and filters marginalize older states and absorb the corresponding information in a Gaussian prior.

The second dimension regards the representation of the uncertainty for the measurements and the Gaussian priors: the *Extended Kalman Filter* (EKF) represents the uncertainty using a covariance matrix; instead, *information filters* and smoothers resort to the information matrix (the inverse of the covariance) or the square-root of the

information matrix [Kaess et al., 2012, Wu et al., 2015].

Finally, the third dimension distinguishes existing approaches by looking at the number of times in which the measurement model is linearized. While a standard EKF (in contrast to the *iterated* EKF) processes a measurement only once, a smoothing approach allows linearizing multiple times.

While the terminology is vast, the underlying algorithms are tightly related. For instance, it can be shown that the iterated Extended Kalman filter equations are equivalent to the Gauss-Newton algorithm, commonly used for smoothing [Bell and Cathey, 1993].

Filtering

Filtering algorithms enable efficient estimation by restricting the inference process to the latest state of the system. The complexity of the EKF grows quadratically in the number of estimated landmarks, therefore, a small number of landmarks (in the order of 20) are typically tracked to allow real-time operation [Davison et al., 2007, Bloesch et al., 2015, Jones and Soatto, 2011]. An alternative is to adopt a "structureless" approach where landmark positions are marginalized out of the state vector. An elegant example of this strategy is the *Multi-State Constraint Kalman filter* (MSC-KF) [Mourikis and Roumeliotis, 2007]. The structureless approach requires to keep previous poses in the state vector, by means of *stochastic cloning* [Roumeliotis and Burdick, 2002].

A drawback of using a structureless approach for filtering, is that the processing of landmark measurements needs to be delayed until all measurements of a landmark are obtained [Mourikis and Roumeliotis, 2007]. This hinders accuracy as the filter cannot use all current visual information. Marginalization is also a source of errors as it locks in linearization errors and erroneous outlier measurements. Therefore, it is particularly important to filter out spurious measurements as a single outlier can irreversibly corrupt the filter [Tsotsos et al., 2015]. Further, linearization errors introduce drift in the estimate and render the filter *inconsistent*. An effect of inconsistency is that the estimator becomes over-confident, resulting in non-optimal information fusion. Generally, the VINproblem has four unobservable directions: the global position and the orientation around the gravity direction (yaw) [Martinelli, 2013, Kottas et al., 2012]. In [Kottas et al., 2012] it is shown that linearization at the wrong estimate results in only three unobservable directions (the global position); hence, erroneous linearization adds spurious information in yaw direction to the Gaussian prior, which renders the filter inconsistent. This problem was addressed with the *first-estimates jacobian* approach [Huang et al., 2008], which ensures that a state is not updated with different linearization points — a source of inconsistency. In the *observability-constrained* EKF (OC-EKF) an estimate of the unobservable directions is maintained which allows to

update the filter only in directions that are observable [Kottas et al., 2012, Hesch et al., 2014]. A thorough analysis of VINobservability properties is given in [Martinelli, 2012, 2013, Hernandez et al., 2015].

Fixed-lag Smoothing

Fixed-lag smoothers estimate the states that fall within a given time window, while marginalizing out older states [Mourikis and Roumeliotis, 2008, Sibley et al., 2010, Dong-Si and Mourikis, 2011, Leutenegger et al., 2013, 2015]. In a maximum likelihood estimation setup, fixed-lag smoothers lead to an optimization problem over a set of recent states. For nonlinear problems, smoothing approaches are generally more accurate than filtering, since they relinearize past measurements [Maybeck, 1979]. Moreover, these approaches are more resilient to outliers, which can be discarded a posteriori (i.e., after the optimization), or can be alleviated by using robust cost functions. On the downside, the marginalization of the states outside the estimation window leads to dense Gaussian priors which hinder efficient inference. For this reason, it has been proposed to drop certain measurements in the interest of sparsity [Leutenegger et al., 2015]. Furthermore, due to marginalization, fixed-lag smoothers share part of the issues of filtering (consistency, build-up of linearization errors) [Huang et al., 2011b, Dong-Si and Mourikis, 2011, Hesch et al., 2014].

Full Smoothing

Full smoothing methods estimate the entire history of the states (camera trajectory and 3D landmarks), by solving a large nonlinear optimization problem [Jung and Taylor, 2001, Sterlow and Singh, 2004, Bryson et al., 2009, Indelman et al., 2013b, Patron-Perez et al., 2015]. Full smoothing guarantees the highest accuracy; however, real-time operation quickly becomes infeasible as the trajectory and the map grow over time. Therefore, it has been proposed to discard frames except selected *keyframes* [Strasdat et al., 2010, Klein and Murray, 2009, Nerurkar et al., 2014, Leutenegger et al., 2015] or to run the optimization in a parallel thread, using a tracking and mapping dual architecture [Klein and Murray, 2007, Mourikis and Roumeliotis, 2008]. A breakthrough has been the development of *incremental smoothing techniques* (iSAM [Kaess et al., 2008], iSAM2 [Kaess et al., 2012]), which leverage the expressiveness of *factor graphs* to maintain sparsity and to identify and update only the typically small subset of variables affected by a new measurement.

Nevertheless, the high rate of inertial measurements (usually 100 Hz to 1 kHz) still constitutes a challenge for smoothing approaches. A naive implementation would require adding a new state at every IMU measurement, which quickly becomes impractically slow [Indelman et al., 2012]. Therefore, inertial measurements are typically

integrated between frames to form relative motion constraints [Indelman et al., 2013a,b, Shen, 2014, Keivan et al., 2014, Leutenegger et al., 2015]. For standard IMU integration between two frames, the initial condition is given by the state estimate at the first frame. However, at every iteration of the optimization, the state estimate changes, which requires to repeat the IMU integration between all frames [Leutenegger et al., 2015]. Lupton and Sukkarieh [2012] show that this repeated integration can be avoided by a reparametrization of the relative motion constraints. Such reparametrization is called *IMU preintegration*.

In the present work, we build upon the seminal work [Lupton and Sukkarieh, 2012] and bring the theory of IMU preintegration to maturity by properly addressing the manifold structure of the rotation group $\mathrm{SO}(3)$. The work [Lupton and Sukkarieh, 2012] adopted Euler angles as global parametrization for rotations. Using Euler angles and applying the usual averaging and smoothing techniques of Euclidean spaces for state propagation and covariance estimation is not properly invariant under the action of rigid transformations [Hornegger and Tomasi, 1999, Moakher, 2002]. Moreover, Euler angles are known to have singularities. Our work, on the other hand, provides a formal treatment of the rotation measurements (and the corresponding noise), and provides a complete derivation of the maximum a posteriori estimator. We also derive analytic expressions for the Jacobians (needed for the optimization), which, to the best of our knowledge, have not been previously reported in the literature. In the experimental section, we show that a proper representation of the rotation manifold results in higher accuracy and robustness, leading to tangible advantages over the original proposal [Lupton and Sukkarieh, 2012].

Preliminaries

In this paper we formulate VINin terms of MAP estimation. In our model, MAP estimation leads to a nonlinear optimization problem that involves quantities living on smooth manifolds (e.g., rotations, poses). Therefore, before delving into details, we conveniently review some useful geometric concepts. This section can be skipped by the expert reader.

We structure this section as follows: Section C.3.1 provides useful notions related to two main Riemannian manifolds: the Special Orthogonal Group $\mathrm{SO}(3)$and the Special Euclidean Group $\mathrm{SE}(3)$. Our presentation is based on [Chirikjian, 2012, Wang and Chirikjian, 2008]. Section C.3.2 describes a suitable model to describe uncertain rotations in $\mathrm{SO}(3)$. Section C.3.3 reviews optimization on manifolds, following standard references [Absil et al., 2007].

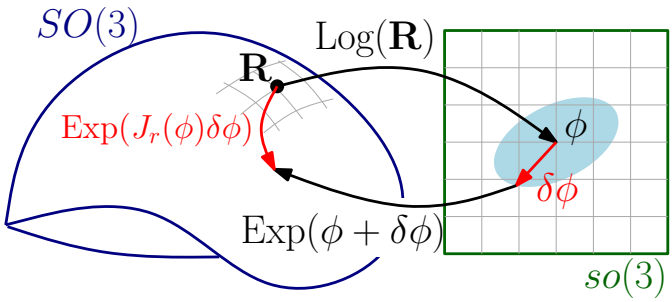


Figure C.1 – The right Jacobian J_r relates an additive perturbation $\delta\boldsymbol{\phi}$ in the tangent space to a multiplicative perturbation on the manifold $\mathrm{SO}(3)$, as per Eq. (C.7).

Notions of Riemannian geometry

Special Orthogonal Group $\mathrm{SO}(3)$describes the group of 3D rotation matrices and it is formally defined as $\mathrm{SO}(3) \doteq \{\mathtt{R} \in \mathbb{R}^{3\times 3} : \mathtt{R}^{\mathsf{T}}\mathtt{R} = \mathbf{I}, \det(\mathtt{R}) = 1\}$. The group operation is the usual matrix multiplication, and the inverse is the matrix transpose. The group $\mathrm{SO}(3)$also forms a smooth manifold. The tangent space to the manifold (at the identity) is denoted as $\mathfrak{so}(3)$, which is also called the *Lie algebra* and coincides with the space of 3×3 skew symmetric matrices. We can identify every skew symmetric matrix with a vector in $\mathbb{R}^3$ using the *hat* operator:

$$\boldsymbol{\omega}^{\wedge} = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix}^{\wedge} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \in \mathfrak{so}(3). \tag{C.1}$$

Similarly, we can map a skew symmetric matrix to a vector in $\mathbb{R}^3$ using the *vee* operator $(\cdot)^{\vee}$: for a skew symmetric matrix $\mathtt{S} = \boldsymbol{\omega}^{\wedge}$, the vee operator is such that $\mathtt{S}^{\vee} = \boldsymbol{\omega}$. A property of skew symmetric matrices that will be useful later on is:

$$\mathbf{a}^{\wedge}\,\mathbf{b} = -\mathbf{b}^{\wedge}\,\mathbf{a}, \quad \forall\, \mathbf{a}, \mathbf{b} \in \mathbb{R}^3. \tag{C.2}$$

The *exponential map* (at the identity) $\exp : \mathfrak{so}(3) \to \mathrm{SO}(3)$ associates an element of the Lie Algebra to a rotation and coincides with standard matrix exponential (Rodrigues' formula):

$$\exp(\boldsymbol{\phi}^{\wedge}) = \mathbf{I} + \tfrac{\sin(\|\boldsymbol{\phi}\|)}{\|\boldsymbol{\phi}\|}\boldsymbol{\phi}^{\wedge} + \tfrac{1-\cos(\|\boldsymbol{\phi}\|)}{\|\boldsymbol{\phi}\|^2}\left(\boldsymbol{\phi}^{\wedge}\right)^2. \tag{C.3}$$

A first-order approximation of the exponential map that we will use later on is:

$$\exp(\boldsymbol{\phi}^{\wedge}) \approx \mathbf{I} + \boldsymbol{\phi}^{\wedge}. \tag{C.4}$$

The *logarithm map* (at the identity) associates a matrix $\mathtt{R} \neq \mathbf{I}$ in $\mathrm{SO}(3)$ to a skew

symmetric matrix:

$$\log(\mathtt{R}) = \frac{\varphi \cdot (\mathtt{R} - \mathtt{R}^\mathsf{T})}{2\sin(\varphi)} \text{ with } \varphi = \cos^{-1}\left(\frac{\operatorname{tr}(\mathtt{R}) - 1}{2}\right). \tag{C.5}$$

Note that $\log(\mathtt{R})^\vee = \mathbf{a}\varphi$, where $\mathbf{a}$ and φ are the rotation axis and the rotation angle of $\mathtt{R}$, respectively. If $\mathtt{R} = \mathbf{I}$, then $\varphi = 0$ and $\mathbf{a}$ is undetermined and can therefore be chosen arbitrarily.

The exponential map is a bijection if restricted to the open ball $\|\boldsymbol{\phi}\| < \pi$, and the corresponding inverse is the logarithm map. However, if we do not restrict the domain, the exponential map becomes surjective as every vector $\boldsymbol{\phi} = (\varphi + 2k\pi)\mathbf{a}$, $k \in \mathbb{Z}$ would be an admissible logarithm of $\mathtt{R}$.

For notational convenience, we adopt "vectorized" versions of the exponential and logarithm map:

$$\begin{array}{rccccccc} \mathrm{Exp}: & \mathbb{R}^3 & \to & \mathrm{SO}(3) & ; & \boldsymbol{\phi} & \mapsto & \exp(\boldsymbol{\phi}^\wedge) \\ \mathrm{Log}: & \mathrm{SO}(3) & \to & \mathbb{R}^3 & ; & \mathtt{R} & \mapsto & \log(\mathtt{R})^\vee, \end{array} \tag{C.6}$$

which operate directly on vectors, rather than on skew symmetric matrices in $\mathfrak{so}(3)$.

Later, we will use the following first-order approximation:

$$\mathrm{Exp}(\boldsymbol{\phi} + \delta\boldsymbol{\phi}) \approx \mathrm{Exp}(\boldsymbol{\phi})\,\mathrm{Exp}(\mathtt{J}_r(\boldsymbol{\phi})\delta\boldsymbol{\phi}). \tag{C.7}$$

The term $\mathtt{J}_r(\boldsymbol{\phi})$ is the *right Jacobian* of $\mathrm{SO}(3)$ [Chirikjian, 2012, p.40] and relates additive increments in the tangent space to multiplicative increments applied on the right-hand-side (Fig. C.1):

$$\mathtt{J}_r(\boldsymbol{\phi}) = \mathbf{I} - \frac{1-\cos(\|\boldsymbol{\phi}\|)}{\|\boldsymbol{\phi}\|^2}\boldsymbol{\phi}^\wedge + \frac{\|\boldsymbol{\phi}\| - \sin(\|\boldsymbol{\phi}\|)}{\|\boldsymbol{\phi}^3\|}(\boldsymbol{\phi}^\wedge)^2. \tag{C.8}$$

A similar first-order approximation holds for the logarithm:

$$\mathrm{Log}\Big(\,\mathrm{Exp}(\boldsymbol{\phi})\,\mathrm{Exp}(\delta\boldsymbol{\phi})\,\Big) \approx \boldsymbol{\phi} + \mathtt{J}_r^{-1}(\boldsymbol{\phi})\delta\boldsymbol{\phi}. \tag{C.9}$$

Where the inverse of the right Jacobian is

$$\mathtt{J}_r^{-1}(\boldsymbol{\phi}) = \mathbf{I} + \frac{1}{2}\boldsymbol{\phi}^\wedge + \left(\frac{1}{\|\boldsymbol{\phi}\|^2} + \frac{1+\cos(\|\boldsymbol{\phi}\|)}{2\|\boldsymbol{\phi}\|\sin(\|\boldsymbol{\phi}\|)}\right)(\boldsymbol{\phi}^\wedge)^2.$$

The right Jacobian $\mathtt{J}_r(\boldsymbol{\phi})$ and its inverse $\mathtt{J}_r^{-1}(\boldsymbol{\phi})$ reduce to the identity matrix for $\|\boldsymbol{\phi}\| = 0$.

Another useful property of the exponential map is:

$$\mathtt{R}\,\mathrm{Exp}(\boldsymbol{\phi})\,\mathtt{R}^{\mathsf{T}} = \exp(\mathtt{R}\boldsymbol{\phi}^{\wedge}\mathtt{R}^{\mathsf{T}}) = \mathrm{Exp}(\mathtt{R}\boldsymbol{\phi}) \tag{C.10}$$

$$\Leftrightarrow \quad \mathrm{Exp}(\boldsymbol{\phi})\,\mathtt{R} = \mathtt{R}\,\mathrm{Exp}(\mathtt{R}^{\mathsf{T}}\boldsymbol{\phi}). \tag{C.11}$$

Special Euclidean Group $\mathrm{SE}(3)$describes the group of rigid motion in 3D, which is the semi-direct product of $\mathrm{SO}(3)$ and $\mathbb{R}^3$, and it is defined as $\mathrm{SE}(3) \doteq \{(\mathtt{R}, \mathbf{p}) : \mathtt{R} \in \mathrm{SO}(3), \mathbf{p} \in \mathbb{R}^3\}$. Given $\mathtt{T}_1, \mathtt{T}_2 \in \mathrm{SE}(3)$, the group operation is $\mathtt{T}_1 \cdot \mathtt{T}_2 = (\mathtt{R}_1\mathtt{R}_2\ ,\ \mathbf{p}_1 + \mathtt{R}_1\mathbf{p}_2)$, and the inverse is $\mathtt{T}_1^{-1} = (\mathtt{R}_1^{\mathsf{T}}\ ,\ -\mathtt{R}_1^{\mathsf{T}}\mathbf{p}_1)$. The *exponential map* and the *logarithm map* for $\mathrm{SE}(3)$are defined in [Wang and Chirikjian, 2008]. However, these are not needed in this paper for reasons that will be clear in Section C.3.3.

Uncertainty Description in $\mathrm{SO}(3)$

A natural definition of uncertainty in $\mathrm{SO}(3)$ is to define a distribution in the tangent space, and then map it to $\mathrm{SO}(3)$ via the exponential map (C.6) [Barfoot and Furgale, 2014, Wang and Chirikjian, 2006, 2008]:

$$\tilde{\mathtt{R}} = \mathtt{R}\,\mathrm{Exp}(\epsilon), \qquad \epsilon \sim \mathcal{N}(0, \Sigma), \tag{C.12}$$

where $\mathtt{R}$ is a given noise-free rotation (the *mean*) and ϵ is a small normally distributed perturbation with zero mean and covariance Σ.

To obtain an explicit expression for the distribution of $\tilde{\mathtt{R}}$, we start from the integral of the Gaussian distribution in $\mathbb{R}^3$:

$$\int_{\mathbb{R}^3} p(\epsilon) d\epsilon = \int_{\mathbb{R}^3} \alpha e^{-\frac{1}{2}\|\epsilon\|_{\Sigma}^2} \mathrm{d}\epsilon = 1, \tag{C.13}$$

where $\alpha = 1/\sqrt{(2\pi)^3 \det(\Sigma)}$ and $\|\epsilon\|_{\Sigma}^2 \doteq \epsilon^{\mathsf{T}}\Sigma^{-1}\epsilon$ is the squared Mahalanobis distance with covariance Σ. Then, applying the change of coordinates $\epsilon = \mathrm{Log}(\mathtt{R}^{-1}\tilde{\mathtt{R}})$ (this is the inverse of (C.12) when $\|\epsilon\| < \pi$), the integral (C.13) becomes:

$$\int_{\mathrm{SO}(3)} \beta(\tilde{\mathtt{R}})\, e^{-\frac{1}{2}\left\|\mathrm{Log}(\mathtt{R}^{-1}\tilde{\mathtt{R}})\right\|_{\Sigma}^2} \mathrm{d}\tilde{\mathtt{R}} = 1, \tag{C.14}$$

where $\beta(\tilde{\mathtt{R}})$ is a normalization factor. The normalization factor assumes the form $\beta(\tilde{\mathtt{R}}) = \alpha/|\det(\mathcal{J}(\tilde{\mathtt{R}})|$, where $\mathcal{J}(\tilde{\mathtt{R}}) \doteq \mathtt{J}_r(\mathrm{Log}(\mathtt{R}^{-1}\tilde{\mathtt{R}}))$ and $\mathtt{J}_r(\cdot)$ is the right Jacobian (C.8); $\mathcal{J}(\tilde{\mathtt{R}})$ is a by-product of the change of variables, see [Barfoot and Furgale, 2014] for a derivation.

From the argument of (C.14) we can directly read our "Gaussian" distribution in $\mathrm{SO}(3)$:

$$p(\tilde{\mathtt{R}}) = \beta(\tilde{\mathtt{R}})\, e^{-\frac{1}{2}\left\|\mathrm{Log}(\mathtt{R}^{-1}\tilde{\mathtt{R}})\right\|_{\Sigma}^{2}}. \tag{C.15}$$

For small covariances we can approximate $\beta \simeq \alpha$, as $\mathtt{J}_r(\mathrm{Log}(\mathtt{R}^{-1}\tilde{\mathtt{R}}))$ is well approximated by the identity matrix when $\tilde{\mathtt{R}}$ is close to $\mathtt{R}$. Note that (C.14) already assumes relatively a small covariance Σ, since it "clips" the probability tails outside the open ball of radius π (this is due to the re-parametrization $\epsilon = \mathrm{Log}(\mathtt{R}^{-1}\tilde{\mathtt{R}})$, which restricts ϵ to $\|\epsilon\| < \pi$). Approximating β as a constant, the negative log-likelihood of a rotation $\mathtt{R}$, given a measurement $\tilde{\mathtt{R}}$ distributed as in (C.15), is:

$$\mathcal{L}(\mathtt{R}) = \frac{1}{2}\left\|\mathrm{Log}(\mathtt{R}^{-1}\tilde{\mathtt{R}})\right\|_{\Sigma}^{2} + \text{const} = \frac{1}{2}\left\|\mathrm{Log}(\tilde{\mathtt{R}}^{-1}\mathtt{R})\right\|_{\Sigma}^{2} + \text{const}, \tag{C.16}$$

which geometrically can be interpreted as the squared angle (geodesic distance in $\mathrm{SO}(3)$) between $\tilde{\mathtt{R}}$ and $\mathtt{R}$ weighted by the inverse uncertainty Σ^{-1}.

Gauss-Newton Method on Manifold

A standard Gauss-Newton method in Euclidean space works by repeatedly optimizing a quadratic approximation of the (generally non-convex) objective function. Solving the quadratic approximation reduces to solving a set of linear equations (*normal equations*), and the solution of this local approximation is used to update the current estimate. Here we recall how to extend this approach to (unconstrained) optimization problems whose variables belong to some manifold $\mathcal{M}$.

Let us consider the following optimization problem:

$$\min_{x \in \mathcal{M}} f(x), \tag{C.17}$$

where the variable x belongs to a manifold $\mathcal{M}$; for the sake of simplicity we consider a single variable in (C.17), while the description easily generalizes to multiple variables.

Contrarily to the Euclidean case, one cannot directly approximate (C.17) as a quadratic function of x. This is due to two main reasons. First, working directly on x leads to an over-parametrization of the problem (e.g., we parametrize a rotation matrix with 9 elements, while a 3D rotation is completely defined by a vector in $\mathbb{R}^3$) and this can make the *normal equations* under-determined. Second, the solution of the resulting approximation does not belong to $\mathcal{M}$ in general.

A standard approach for optimization on manifold [Absil et al., 2007, Smith, 1994], consists of defining a *retraction* $\mathcal{R}_x$, which is a bijective map between an element δx of the tangent space (at x) and a neighborhood of $x \in \mathcal{M}$. Using the retraction, we can

re-parametrize our problem as follows:

$$\min_{x \in \mathcal{M}} f(x) \quad \Rightarrow \quad \min_{\delta x \in \mathbb{R}^n} f(\mathcal{R}_x(\delta x)). \tag{C.18}$$

The re-parametrization is usually called *lifting* [Absil et al., 2007]. Roughly speaking, we work in the tangent space defined at the current estimate, which locally behaves as an Euclidean space. The use of the retraction allows framing the optimization problem over an Euclidean space of suitable dimension (e.g., $\delta x \in \mathbb{R}^3$ when we work in $\mathrm{SO}(3)$). We can now apply standard optimization techniques to the problem on the right-hand side of (C.18). In the Gauss-Newton framework, we square the cost around the current estimate. Then we solve the quadratic approximation to get a vector $\delta x^\star$ in the tangent space. Finally, the current guess on the manifold is updated as

$$\hat{x} \leftarrow \mathcal{R}_{\hat{x}}(\delta x^\star). \tag{C.19}$$

This "lift-solve-retract" scheme can be generalized to any trust-region method [Absil et al., 2007]. Moreover, it provides a grounded and unifying generalization of the *error state model*, commonly used in aerospace literature for filtering [Farrell, 2008] and recently adopted in robotics for optimization [Leutenegger et al., 2013, Nerurkar et al., 2014].

We conclude this section by discussing the choice of the retraction $\mathcal{R}_x$. A possible retraction is the exponential map. It is known that, computationally, this may not be the most convenient choice, see [Manton, 2002].

In this work, we use the following retraction for $\mathrm{SO}(3)$,

$$\mathcal{R}_{\mathtt{R}}(\boldsymbol{\phi}) = \mathtt{R}\,\mathrm{Exp}(\delta\boldsymbol{\phi}), \qquad \delta\boldsymbol{\phi} \in \mathbb{R}^3, \tag{C.20}$$

and for $\mathrm{SE}(3)$, we use the retraction at $\mathtt{T} \doteq (\mathtt{R}, \mathbf{p})$:

$$\mathcal{R}_{\mathtt{T}}(\delta\boldsymbol{\phi}, \delta\mathbf{p}) = (\mathtt{R}\,\mathrm{Exp}(\delta\boldsymbol{\phi}),\ \mathbf{p} + \mathtt{R}\,\delta\mathbf{p}), \qquad [\delta\boldsymbol{\phi} \ \ \delta\mathbf{p}] \in \mathbb{R}^6, \tag{C.21}$$

which explains why in Section C.3.1 we only defined the exponential map for $\mathrm{SO}(3)$: with this choice of retraction we never need to compute the exponential map for $\mathrm{SE}(3)$.

Maximum a Posteriori Visual-Inertial State Estimation

We consider a VINproblem in which we want to track the state of a *sensing system* (e.g., a mobile robot, a UAV, or a hand-held device), equipped with an IMU and a monocular camera. We assume that the IMU frame "B" coincides with the body frame we want to track, and that the transformation between the camera and the IMU is fixed

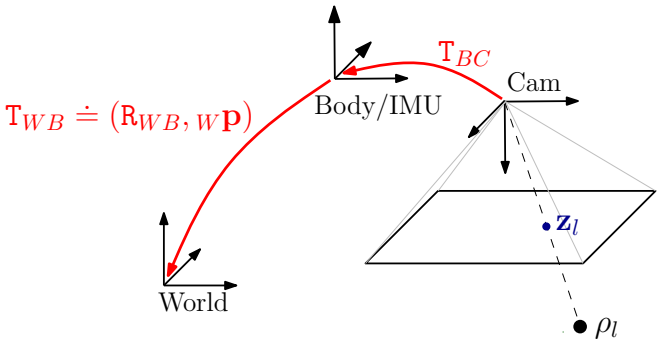


Figure C.2 – $\mathtt{T}_{WB} \doteq (\mathtt{R}_{WB}, {}_W\mathbf{p})$ is the pose of the body frame B w.r.t. the world frame W. We assume that the body frame coincides with the IMU frame. $\mathtt{T}_{BC}$ is the pose of the camera in the body frame, known from prior calibration.

and known from prior calibration (Fig. C.2). Furthermore, we assume that a *front-end* provides image measurements of 3D landmarks at unknown position. The front-end also selects a subset of images, called *keyframes* [Strasdat et al., 2010], for which we want to compute a pose estimate. Section C.8.2 discusses implementation aspects, including the choice of the front-end in our experiments.

The State

The state of the system at time i is described by the IMU orientation, position, velocity and biases:

$$\mathbf{x}_i \doteq [\mathtt{R}_i, \mathbf{p}_i, \mathbf{v}_i, \mathbf{b}_i]. \tag{C.22}$$

The pose $(\mathtt{R}_i, \mathbf{p}_i)$ belongs to SE(3), while velocities live in a vector space, i.e., $\mathbf{v}_i \in \mathbb{R}^3$. IMU biases can be written as $\mathbf{b}_i = [\mathbf{b}_i^g \;\; \mathbf{b}_i^a] \in \mathbb{R}^6$, where $\mathbf{b}_i^g, \mathbf{b}_i^a \in \mathbb{R}^3$ are the gyroscope and accelerometer bias, respectively.

Let $\mathcal{K}_k$ denote the set of all keyframes up to time k. In our approach we estimate the state of all keyframes:

$$\mathcal{X}_k \doteq \{\mathbf{x}_i\}_{i \in \mathcal{K}_k}. \tag{C.23}$$

In our implementation, we adopt a structureless approach (*cf.*, Section C.7), hence the 3D landmarks are not part of the variables to be estimated. However, the proposed approach generalizes in a straightforward manner to also estimating the landmarks and the camera intrinsic and extrinsic calibration parameters.

The Measurements

The input to our estimation problem are the measurements from the camera and the IMU. We denote with $\mathcal{C}_i$ the image measurements at keyframe i. At time i, the camera can observe multiple landmarks l, hence $\mathcal{C}_i$ contains multiple image measurements $\mathbf{z}_{il}$. With slight abuse of notation we write $l \in \mathcal{C}_i$ when a landmark l is seen at time i.

We denote with $\mathcal{I}_{ij}$ the set of IMU measurements acquired between two consecutive keyframes i and j. Depending on the IMU measurement rate and the frequency of selected keyframes, each set $\mathcal{I}_{ij}$ can contain from a small number to hundreds of IMU measurements. The set of measurements collected up to time k is

$$\mathcal{Z}_k \doteq \{\mathcal{C}_i, \mathcal{I}_{ij}\}_{(i,j)\in\mathcal{K}_k}. \tag{C.24}$$

Factor Graphs and MAP Estimation

The posterior probability of the variables $\mathcal{X}_k$, given the available visual and inertial measurements $\mathcal{Z}_k$ and priors $p(\mathcal{X}_0)$ is:

$$\begin{aligned} p(\mathcal{X}_k|\mathcal{Z}_k) \propto\ & p(\mathcal{X}_0)p(\mathcal{Z}_k|\mathcal{X}_k) \overset{(a)}{=} p(\mathcal{X}_0)\prod_{(i,j)\in\mathcal{K}_k} p(\mathcal{C}_i, \mathcal{I}_{ij}|\mathcal{X}_k) \\ & \overset{(b)}{=} p(\mathcal{X}_0)\prod_{(i,j)\in\mathcal{K}_k} p(\mathcal{I}_{ij}|\mathbf{x}_i, \mathbf{x}_j) \prod_{i\in\mathcal{K}_k}\prod_{l\in\mathcal{C}_i} p(\mathbf{z}_{il}|\mathbf{x}_i). \end{aligned} \tag{C.25}$$

The factorizations (a) and (b) follow from a standard independence assumption among the measurements. Furthermore, the Markovian property is applied in (b) (e.g., an image measurement at time i only depends on the state at time i).

As the measurements $\mathcal{Z}_k$ are known, we are free to eliminate them as variables and consider them as parameters of the joint probability factors over the actual unknowns. This naturally leads to the well known factor graph representation, a class of bipartite graphical models that can be used to represent such factored densities [Kschischang et al., 2001, Dellaert, 2005]. A schematic representation of the connectivity of the factor graph underlying the VINproblem is given in Fig. C.3 (the connectivity of the structureless vision factors will be clarified in Section C.7). The factor graph is composed of nodes for unknowns and nodes for the probability factors defined on them, and the graph structure expresses which unknowns are involved in each factor.

The MAPestimate $\mathcal{X}_k^\star$ corresponds to the maximum of (C.25), or equivalently, the minimum of the negative log-posterior. Under the assumption of zero-mean Gaussian

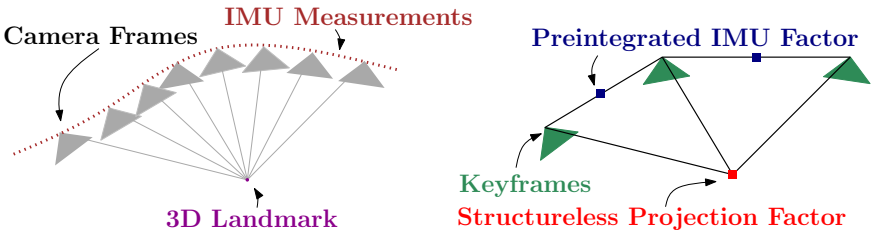


Figure C.3 – Left: visual and inertial measurements in VIN. Right: factor graph in which several IMU measurements are summarized in a single preintegrated IMU factor and a structureless vision factor constraints keyframes observing the same landmark.

noise, the negative log-posterior can be written as a sum of squared residual errors:

$$\begin{aligned}\mathcal{X}_k^\star &\doteq \arg\min_{\mathcal{X}_k} \; -\log_e \; p(\mathcal{X}_k|\mathcal{Z}_k) \\ &= \arg\min_{\mathcal{X}_k} \; \|\mathbf{r}_0\|^2_{\mathbf{\Sigma}_0} + \sum_{(i,j)\in\mathcal{K}_k} \|\mathbf{r}_{\mathcal{I}_{ij}}\|^2_{\mathbf{\Sigma}_{ij}} + \sum_{i\in\mathcal{K}_k}\sum_{l\in\mathcal{C}_i} \|\mathbf{r}_{\mathcal{C}_{il}}\|^2_{\mathbf{\Sigma}_\mathcal{C}}\end{aligned} \tag{C.26}$$

where $\mathbf{r}_0$, $\mathbf{r}_{\mathcal{I}_{ij}}$, $\mathbf{r}_{\mathcal{C}_{il}}$ are the residual errors associated to the measurements, and $\mathbf{\Sigma}_0$, $\mathbf{\Sigma}_{ij}$, and $\mathbf{\Sigma}_\mathcal{C}$ are the corresponding covariance matrices. Roughly speaking, the residual error is a function of $\mathcal{X}_k$ that quantifies the mismatch between a measured quantity and the predicted value of this quantity given the state $\mathcal{X}_k$ and the priors. The goal of the following sections is to provide expressions for the residual errors and the covariances.

IMU Model and Motion Integration

An IMU commonly includes a 3-axis accelerometer and a 3-axis gyroscope and allows measuring the rotation rate and the acceleration of the sensor with respect to an inertial frame. The measurements, namely ${}_\mathrm{B}\tilde{\mathbf{a}}(t)$, and ${}_\mathrm{B}\tilde{\boldsymbol{\omega}}_\mathrm{WB}(t)$, are affected by additive white noise $\boldsymbol{\eta}$ and a slowly varying sensor bias $\mathbf{b}$:

$${}_\mathrm{B}\tilde{\boldsymbol{\omega}}_\mathrm{WB}(t) = {}_\mathrm{B}\boldsymbol{\omega}_\mathrm{WB}(t) + \mathbf{b}^g(t) + \boldsymbol{\eta}^g(t) \tag{C.27}$$

$${}_\mathrm{B}\tilde{\mathbf{a}}(t) = \mathrm{R}^\mathsf{T}_\mathrm{WB}(t)\,({}_\mathrm{W}\mathbf{a}(t) - {}_\mathrm{W}\mathbf{g}) + \mathbf{b}^a(t) + \boldsymbol{\eta}^a(t), \tag{C.28}$$

In our notation, the prefix B denotes that the corresponding quantity is expressed in the frame B (*c.f.*, Fig. C.2). The pose of the IMU is described by the transformation $\{\mathrm{R}_\mathrm{WB}, {}_\mathrm{W}\mathbf{p}\}$, which maps a point from sensor frame B to W. The vector ${}_\mathrm{B}\boldsymbol{\omega}_\mathrm{WB}(t) \in \mathbb{R}^3$ is the instantaneous angular velocity of B relative to W expressed in coordinate frame B, while ${}_\mathrm{W}\mathbf{a}(t) \in \mathbb{R}^3$ is the acceleration of the sensor; ${}_\mathrm{W}\mathbf{g}$ is the gravity vector in world coordinates. We neglect effects due to earth's rotation, which amounts to assuming that W is an inertial frame.

The goal now is to infer the motion of the system from IMU measurements. For this

purpose we introduce the following kinematic model [Murray et al., 1994, Farrell, 2008]:

$$\dot{\mathrm{R}}_{\mathrm{WB}} = \mathrm{R}_{\mathrm{WB}}\,{}_{\mathrm{B}}\boldsymbol{\omega}^{\wedge}_{\mathrm{WB}}, \qquad {}_{\mathrm{W}}\dot{\mathbf{v}} = {}_{\mathrm{W}}\mathbf{a}, \qquad {}_{\mathrm{W}}\dot{\mathbf{p}} = {}_{\mathrm{W}}\mathbf{v}, \tag{C.29}$$

which describes the evolution of the pose and velocity of B.

The state at time $t + \Delta t$ is obtained by integrating Eq. (C.29):

$$\begin{aligned}
\mathrm{R}_{\mathrm{WB}}(t+\Delta t) &= \mathrm{R}_{\mathrm{WB}}(t)\,\mathrm{Exp}\left(\int_t^{t+\Delta t} {}_{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{WB}}(\tau)d\tau\right) \\
{}_{\mathrm{W}}\mathbf{v}(t+\Delta t) &= {}_{\mathrm{W}}\mathbf{v}(t) + \int_t^{t+\Delta t} {}_{\mathrm{W}}\mathbf{a}(\tau)d\tau \\
{}_{\mathrm{W}}\mathbf{p}(t+\Delta t) &= {}_{\mathrm{W}}\mathbf{p}(t) + \int_t^{t+\Delta t} {}_{\mathrm{W}}\mathbf{v}(\tau)d\tau + \iint_t^{t+\Delta t} {}_{\mathrm{W}}\mathbf{a}(\tau)d\tau^2.
\end{aligned}$$

Assuming that ${}_{\mathrm{W}}\mathbf{a}$ and ${}_{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{WB}}$ remain constant in the time interval $[t, t+\Delta t]$, we can write:

$$\begin{aligned}
\mathrm{R}_{\mathrm{WB}}(t+\Delta t) &= \mathrm{R}_{\mathrm{WB}}(t)\,\mathrm{Exp}\left({}_{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{WB}}(t)\Delta t\right) \\
{}_{\mathrm{W}}\mathbf{v}(t+\Delta t) &= {}_{\mathrm{W}}\mathbf{v}(t) + {}_{\mathrm{W}}\mathbf{a}(t)\Delta t \\
{}_{\mathrm{W}}\mathbf{p}(t+\Delta t) &= {}_{\mathrm{W}}\mathbf{p}(t) + {}_{\mathrm{W}}\mathbf{v}(t)\Delta t + \frac{1}{2}{}_{\mathrm{W}}\mathbf{a}(t)\Delta t^2.
\end{aligned} \tag{C.30}$$

Using Eqs. (C.27)–(C.28), we can write ${}_{\mathrm{W}}\mathbf{a}$ and ${}_{\mathrm{B}}\boldsymbol{\omega}_{\mathrm{WB}}$ as a function of the IMU measurements, hence (C.30) becomes

$$\begin{aligned}
\mathrm{R}(t+\Delta t) &= \mathrm{R}(t)\,\mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}(t) - \mathbf{b}^{g}(t) - \boldsymbol{\eta}^{gd}(t)\right)\Delta t\right) \\
\mathbf{v}(t+\Delta t) &= \mathbf{v}(t) + \mathbf{g}\Delta t + \mathrm{R}(t)\left(\tilde{\mathbf{a}}(t) - \mathbf{b}^{a}(t) - \boldsymbol{\eta}^{ad}(t)\right)\Delta t \\
\mathbf{p}(t+\Delta t) &= \mathbf{p}(t) + \mathbf{v}(t)\Delta t + \frac{1}{2}\mathbf{g}\Delta t^2 \\
&\quad + \frac{1}{2}\mathrm{R}(t)\left(\tilde{\mathbf{a}}(t) - \mathbf{b}^{a}(t) - \boldsymbol{\eta}^{ad}(t)\right)\Delta t^2,
\end{aligned} \tag{C.31}$$

where we dropped the coordinate frame subscripts for readability (the notation should be unambiguous from now on). This numeric integration of the velocity and position assumes a constant orientation $\mathrm{R}(t)$ for the time of integration between two measurements, which is not an exact solution of the differential equation (C.29) for measurements with non-zero rotation rate. In practice, the use of a high-rate IMU mitigates the effects of this approximation. We adopt the integration scheme (C.31) as it is simple and amenable for modeling and uncertainty propagation. While we show that this integration scheme performs very well in practice, we remark that for slower IMU measurement rates one may consider using higher-order numerical integration methods [Crouch and Grossman, 1993, Munthe-Kaas, 1999, Park and Chung, 2005,

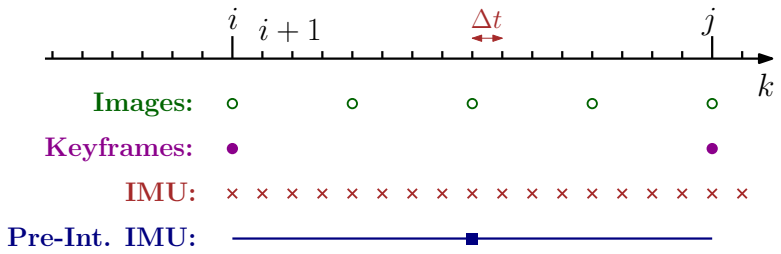


Figure C.4 – Different rates for IMU and camera.

Andrle and Crassidis, 2013].

The covariance of the discrete-time noise $\boldsymbol{\eta}^{gd}$ is a function of the sampling rate and relates to the continuous-time spectral noise $\boldsymbol{\eta}^{g}$ via $\text{Cov}(\boldsymbol{\eta}^{gd}(t)) = \frac{1}{\Delta t}\text{Cov}(\boldsymbol{\eta}^{g}(t))$. The same relation holds for $\boldsymbol{\eta}^{ad}$ (*cf.*, [Crassidis, 2006, Appendix]).

IMU Preintegration on Manifold

While Eq. (C.31) could be readily seen as a probabilistic constraint in a factor graph, it would require to include states in the factor graph at high rate. Intuitively, Eq. (C.31) relates states at time t and $t + \Delta t$, where Δt is the sampling period of the IMU, hence we would have to add new states in the estimation at every new IMU measurement [Indelman et al., 2012].

Here we show that all measurements between two keyframes at times $k = i$ and $k = j$ (see Fig. C.4) can be summarized in a single compound measurement, named *preintegrated IMU measurement*, which constrains the motion between consecutive keyframes. This concept was first proposed in [Lupton and Sukkarieh, 2012] using Euler angles and we extend it, by developing a suitable theory for preintegration on the manifold $\text{SO}(3)$.

We assume that the IMU is synchronized with the camera and provides measurements at discrete times k (*cf.*, Fig. C.4).[1] Iterating the IMU integration (C.31) for all Δt intervals

[1] We calibrate the IMU-camera delay using the *Kalibr* toolbox [Furgale et al., 2013]. An alternative is to add the delay as a state in the estimation process [Li and Mourikis, 2014].

between two consecutive keyframes at times $k = i$ and $k = j$ (*c.f.*, Fig. C.4), we find:

$$\begin{aligned}
\mathtt{R}_j &= \mathtt{R}_i \prod_{k=i}^{j-1} \mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}_k - \mathbf{b}_k^g - \boldsymbol{\eta}_k^{gd}\right)\Delta t\right), \\
\mathbf{v}_j &= \mathbf{v}_i + \mathbf{g}\Delta t_{ij} + \sum_{k=i}^{j-1} \mathtt{R}_k\left(\tilde{\mathbf{a}}_k - \mathbf{b}_k^a - \boldsymbol{\eta}_k^{ad}\right)\Delta t \\
\mathbf{p}_j &= \mathbf{p}_i + \sum_{k=i}^{j-1}\left[\mathbf{v}_k\Delta t + \frac{1}{2}\mathbf{g}\Delta t^2 + \frac{1}{2}\mathtt{R}_k\left(\tilde{\mathbf{a}}_k - \mathbf{b}_k^a - \boldsymbol{\eta}_k^{ad}\right)\Delta t^2\right]
\end{aligned} \tag{C.32}$$

where we introduced the shorthands $\Delta t_{ij} \doteq \sum_{k=i}^{j-1} \Delta t$ and $(\cdot)_i \doteq (\cdot)(t_i)$ for readability. While Eq. (C.32) already provides an estimate of the motion between time t_i and t_j, it has the drawback that the integration in (C.32) has to be repeated whenever the linearization point at time t_i changes [Leutenegger et al., 2015] (intuitively, a change in the rotation $\mathtt{R}_i$ implies a change in all future rotations $\mathtt{R}_k$, $k = i, \ldots, j-1$, and makes necessary to re-evaluate summations and products in (C.32)).

We want to avoid to recompute the above integration whenever the linearization point at time t_i changes. Therefore, we follow [Lupton and Sukkarieh, 2012] and *define* the following relative motion increments that are independent of the pose and velocity at t_i:

$$\begin{aligned}
\Delta\mathtt{R}_{ij} &\doteq \mathtt{R}_i^\mathsf{T}\mathtt{R}_j = \prod_{k=i}^{j-1} \mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}_k - \mathbf{b}_k^g - \boldsymbol{\eta}_k^{gd}\right)\Delta t\right) \\
\Delta\mathbf{v}_{ij} &\doteq \mathtt{R}_i^\mathsf{T}\left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right) = \sum_{k=i}^{j-1} \Delta\mathtt{R}_{ik}\left(\tilde{\mathbf{a}}_k - \mathbf{b}_k^a - \boldsymbol{\eta}_k^{ad}\right)\Delta t \\
\Delta\mathbf{p}_{ij} &\doteq \mathtt{R}_i^\mathsf{T}\left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \tfrac{1}{2}\textstyle\sum_{k=i}^{j-1}\mathbf{g}\Delta t^2\right) \\
&= \sum_{k=i}^{j-1}\left[\Delta\mathbf{v}_{ik}\Delta t + \frac{1}{2}\Delta\mathtt{R}_{ik}\left(\tilde{\mathbf{a}}_k - \mathbf{b}_k^a - \boldsymbol{\eta}_k^{ad}\right)\Delta t^2\right]
\end{aligned} \tag{C.33}$$

where $\Delta\mathtt{R}_{ik} \doteq \mathtt{R}_i^\mathsf{T}\mathtt{R}_k$ and $\Delta\mathbf{v}_{ik} \doteq \mathtt{R}_i^\mathsf{T}\left(\mathbf{v}_k - \mathbf{v}_i - \mathbf{g}\Delta t_{ik}\right)$. We highlight that, in contrast to the "delta" rotation $\Delta\mathtt{R}_{ij}$, neither $\Delta\mathbf{v}_{ij}$ nor $\Delta\mathbf{p}_{ij}$ correspond to the true *physical* change in velocity and position but are defined in a way that make the right-hand side of (C.33) independent from the state at time i as well as gravitational effects. Indeed, we will be able to compute the right-hand side of (C.33) directly from the inertial measurements between the two keyframes.

Unfortunately, summations and products in (C.33) are still function of the bias estimate. We tackle this problem in two steps. In Section C.6.1, we assume $\mathbf{b}_i$ is known; then, in Section C.6.3 we show how to avoid repeating the integration when the bias estimate changes.

In the rest of the paper, we assume that the bias remains constant between two keyframes:

$$\mathbf{b}_i^g = \mathbf{b}_{i+1}^g = \ldots = \mathbf{b}_{j-1}^g, \quad \mathbf{b}_i^a = \mathbf{b}_{i+1}^a = \ldots = \mathbf{b}_{j-1}^a. \tag{C.34}$$

Preintegrated IMU Measurements

Equation (C.33) relates the states of keyframes i and j (left-hand side) to the measurements (right-hand side). In this sense, it can be already understood as a measurement model. Unfortunately, it has a fairly intricate dependence on the measurement noise and this complicates a direct application of MAP estimation; intuitively, the MAP estimator requires to clearly define the densities (and their log-likelihood) of the measurements. In this section we manipulate (C.33) so to make easier the derivation of the measurement log-likelihood. In practice, we isolate the noise terms of the individual inertial measurements in (C.33). As discussed above, across this section assume that the bias at time t_i is known.

Let us start with the rotation increment $\Delta\mathtt{R}_{ij}$ in (C.33). We use the first-order approximation (C.7) (rotation noise is "small") and rearrange the terms, by "moving" the noise to the end, using the relation (C.11):

$$\begin{aligned}
\Delta\mathtt{R}_{ij} &\overset{\text{eq.(C.7)}}{\simeq} \prod_{k=i}^{j-1} \left[\mathrm{Exp}\left((\tilde{\boldsymbol{\omega}}_k - \mathbf{b}_i^g)\,\Delta t \right) \mathrm{Exp}\left(-\mathtt{J}_r^k\, \boldsymbol{\eta}_k^{gd}\, \Delta t \right) \right] \\
&\overset{\text{eq.(C.11)}}{=} \Delta\tilde{\mathtt{R}}_{ij} \prod_{k=i}^{j-1} \mathrm{Exp}\left(-\Delta\tilde{\mathtt{R}}_{k+1j}^{\mathsf{T}}\, \mathtt{J}_r^k\, \boldsymbol{\eta}_k^{gd}\, \Delta t \right) \\
&\doteq \Delta\tilde{\mathtt{R}}_{ij} \mathrm{Exp}\left(-\delta\boldsymbol{\phi}_{ij} \right)
\end{aligned} \tag{C.35}$$

with $\mathtt{J}_r^k \doteq \mathtt{J}_r^k((\tilde{\boldsymbol{\omega}}_k - \mathbf{b}_i^g)\Delta t)$. In the last line of (C.35), we defined the *preintegrated rotation measurement* $\Delta\tilde{\mathtt{R}}_{ij} \doteq \prod_{k=i}^{j-1} \mathrm{Exp}\left((\tilde{\boldsymbol{\omega}}_k - \mathbf{b}_i^g)\,\Delta t \right)$, and its noise $\delta\boldsymbol{\phi}_{ij}$, which will be further analysed in the next section.

Substituting (C.35) back into the expression of $\Delta\mathbf{v}_{ij}$ in (C.33), using the first-order approximation (C.4) for $\mathrm{Exp}\left(-\delta\boldsymbol{\phi}_{ij}\right)$, and dropping higher-order noise terms, we obtain:

$$\begin{aligned}
\Delta\mathbf{v}_{ij} &\overset{\text{eq.(C.4)}}{\simeq} \sum_{k=i}^{j-1} \Delta\tilde{\mathtt{R}}_{ik}(\mathbf{I} - \delta\boldsymbol{\phi}_{ik}^{\wedge})\,(\tilde{\mathbf{a}}_k - \mathbf{b}_i^a)\,\Delta t - \Delta\tilde{\mathtt{R}}_{ik}\boldsymbol{\eta}_k^{ad}\Delta t \\
&\overset{\text{eq.(C.2)}}{=} \Delta\tilde{\mathbf{v}}_{ij} + \sum_{k=i}^{j-1} \left[\Delta\tilde{\mathtt{R}}_{ik}\,(\tilde{\mathbf{a}}_k - \mathbf{b}_i^a)^{\wedge}\,\delta\boldsymbol{\phi}_{ik}\Delta t - \Delta\tilde{\mathtt{R}}_{ik}\boldsymbol{\eta}_k^{ad}\Delta t \right] \\
&\doteq \Delta\tilde{\mathbf{v}}_{ij} - \delta\mathbf{v}_{ij}
\end{aligned} \tag{C.36}$$

where we defined the *preintegrated velocity measurement* $\Delta\tilde{\mathbf{v}}_{ij} \doteq \sum_{k=i}^{j-1} \Delta\tilde{\mathrm{R}}_{ik}\left(\tilde{\mathbf{a}}_k - \mathbf{b}_i^a\right)\Delta t$ and its noise $\delta\mathbf{v}_{ij}$.

Similarly, substituting (C.35) and (C.36) in the expression of $\Delta\mathbf{p}_{ij}$ in (C.33), and using the first-order approximation (C.4), we obtain:

$$\begin{aligned}\Delta\mathbf{p}_{ij} &\overset{\text{eq.(C.4)}}{\simeq} \sum_{k=i}^{j-1}\left[(\Delta\tilde{\mathbf{v}}_{ik} - \delta\mathbf{v}_{ik})\Delta t + \frac{1}{2}\Delta\tilde{\mathrm{R}}_{ik}(\mathbf{I} - \delta\boldsymbol{\phi}_{ik}^{\wedge})\left(\tilde{\mathbf{a}}_k - \mathbf{b}_i^a\right)\Delta t^2 - \frac{1}{2}\Delta\tilde{\mathrm{R}}_{ik}\boldsymbol{\eta}_k^{ad}\Delta t^2\right] \\ &\overset{\text{eq.(C.2)}}{=} \Delta\tilde{\mathbf{p}}_{ij} + \sum_{k=i}^{j-1}\left[-\delta\mathbf{v}_{ik}\Delta t + \frac{1}{2}\Delta\tilde{\mathrm{R}}_{ik}\left(\tilde{\mathbf{a}}_k - \mathbf{b}_i^a\right)^{\wedge}\delta\boldsymbol{\phi}_{ik}\Delta t^2 - \frac{1}{2}\Delta\tilde{\mathrm{R}}_{ik}\boldsymbol{\eta}_k^{ad}\Delta t^2\right] \\ &\doteq \Delta\tilde{\mathbf{p}}_{ij} - \delta\mathbf{p}_{ij},\end{aligned} \tag{C.37}$$

where we defined the *preintegrated position measurement* $\Delta\tilde{\mathbf{p}}_{ij}$ and its noise $\delta\mathbf{p}_{ij}$.

Substituting the expressions (C.35), (C.36), (C.37) back in the original definition of $\Delta\mathrm{R}_{ij}, \Delta\mathbf{v}_{ij}, \Delta\mathbf{p}_{ij}$ in (C.33), we finally get our *preintegrated measurement model* (remember $\mathrm{Exp}(-\delta\boldsymbol{\phi}_{ij})^{\mathsf{T}} = \mathrm{Exp}(\delta\boldsymbol{\phi}_{ij})$):

$$\begin{aligned}\Delta\tilde{\mathrm{R}}_{ij} &= \mathrm{R}_i^{\mathsf{T}}\mathrm{R}_j\mathrm{Exp}\left(\delta\boldsymbol{\phi}_{ij}\right) \\ \Delta\tilde{\mathbf{v}}_{ij} &= \mathrm{R}_i^{\mathsf{T}}\left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right) + \delta\mathbf{v}_{ij} \\ \Delta\tilde{\mathbf{p}}_{ij} &= \mathrm{R}_i^{\mathsf{T}}\left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2\right) + \delta\mathbf{p}_{ij}\end{aligned} \tag{C.38}$$

where our compound measurements are written as a function of the (to-be-estimated) state "plus" a random noise, described by the random vector $[\delta\boldsymbol{\phi}_{ij}^{\mathsf{T}}, \delta\mathbf{v}_{ij}^{\mathsf{T}}, \delta\mathbf{p}_{ij}^{\mathsf{T}}]^{\mathsf{T}}$.

To wrap-up the discussion in this section, we manipulated the measurement model (C.33) and rewrote it as (C.38). The advantage of Eq. (C.38) is that, for a suitable distribution of the noise, it makes the definition of the log-likelihood straightforward. For instance the (negative) log-likelihood of measurements with zero-mean additive Gaussian noise (last two lines in (C.38)) is a quadratic function. Similarly, if $\delta\boldsymbol{\phi}_{ij}$ is a zero-mean Gaussian noise, we compute the (negative) log-likelihood associated with $\Delta\tilde{\mathrm{R}}_{ij}$. The nature of the noise terms is discussed in the following section.

Noise Propagation

In this section we derive the statistics of the noise vector $[\delta\boldsymbol{\phi}_{ij}^{\mathsf{T}}, \delta\mathbf{v}_{ij}^{\mathsf{T}}, \delta\mathbf{p}_{ij}^{\mathsf{T}}]^{\mathsf{T}}$. While we already observed that it is convenient to approximate the noise vector to be zero-mean Normally distributed, it is of paramount importance to accurately model the noise covariance. Indeed, the noise covariance has a strong influence on the MAP estimator (the inverse noise covariance is used to weight the terms in the optimization (C.26)). In this section, we therefore provide a derivation of the covariance $\boldsymbol{\Sigma}_{ij}$ of the preintegrated

measurements:

$$\boldsymbol{\eta}_{ij}^{\Delta} \doteq [\delta\boldsymbol{\phi}_{ij}^{\mathsf{T}}, \delta\mathbf{v}_{ij}^{\mathsf{T}}, \delta\mathbf{p}_{ij}^{\mathsf{T}}]^{\mathsf{T}} \sim \mathcal{N}(\mathbf{0}_{9\times 1}, \boldsymbol{\Sigma}_{ij}). \tag{C.39}$$

We first consider the preintegrated rotation noise $\delta\boldsymbol{\phi}_{ij}$. Recall from (C.35) that

$$\mathrm{Exp}\left(-\delta\boldsymbol{\phi}_{ij}\right) \doteq \prod_{k=i}^{j-1} \mathrm{Exp}\left(-\Delta\tilde{\mathrm{R}}_{k+1j}^{\mathsf{T}} \mathrm{J}_r^k \, \boldsymbol{\eta}_k^{gd} \, \Delta t\right). \tag{C.40}$$

Taking the Log on both sides and changing signs, we get:

$$\delta\boldsymbol{\phi}_{ij} = -\mathrm{Log}\left(\prod_{k=i}^{j-1} \mathrm{Exp}\left(-\Delta\tilde{\mathrm{R}}_{k+1j}^{\mathsf{T}} \mathrm{J}_r^k \, \boldsymbol{\eta}_k^{gd} \, \Delta t\right)\right). \tag{C.41}$$

Repeated application of the first-order approximation (C.9) (recall that $\boldsymbol{\eta}_k^{gd}$ as well as $\delta\boldsymbol{\phi}_{ij}$ are small rotation noises, hence the right Jacobians are close to the identity) produces:

$$\delta\boldsymbol{\phi}_{ij} \simeq \sum_{k=i}^{j-1} \Delta\tilde{\mathrm{R}}_{k+1j}^{\mathsf{T}} \, \mathrm{J}_r^k \, \boldsymbol{\eta}_k^{gd} \, \Delta t \tag{C.42}$$

Up to first order, the noise $\delta\boldsymbol{\phi}_{ij}$ is zero-mean and Gaussian, as it is a linear combination of zero-mean noise terms $\boldsymbol{\eta}_k^{gd}$. This is desirable, since it brings the rotation measurement model (C.38) exactly in the form (C.12).

Dealing with the noise terms $\delta\mathbf{v}_{ij}$ and $\delta\mathbf{p}_{ij}$ is now easy: these are linear combinations of the acceleration noise $\boldsymbol{\eta}_k^{ad}$ and the preintegrated rotation noise $\delta\boldsymbol{\phi}_{ij}$, hence they are also zero-mean and Gaussian. Simple manipulation leads to:

$$\begin{aligned}
\delta\mathbf{v}_{ij} &\simeq \sum_{k=i}^{j-1} \left[-\Delta\tilde{\mathrm{R}}_{ik} \left(\tilde{\mathbf{a}}_k - \mathbf{b}_i^a\right)^{\wedge} \delta\boldsymbol{\phi}_{ik} \Delta t + \Delta\tilde{\mathrm{R}}_{ik} \boldsymbol{\eta}_k^{ad} \Delta t\right] \\
\delta\mathbf{p}_{ij} &\simeq \sum_{k=i}^{j-1} \left[\delta\mathbf{v}_{ik}\Delta t - \frac{1}{2}\Delta\tilde{\mathrm{R}}_{ik} \left(\tilde{\mathbf{a}}_k - \mathbf{b}_i^a\right)^{\wedge} \delta\boldsymbol{\phi}_{ik} \Delta t^2 + \frac{1}{2}\Delta\tilde{\mathrm{R}}_{ik} \boldsymbol{\eta}_k^{ad} \Delta t^2\right]
\end{aligned} \tag{C.43}$$

where the relations are valid up to the first order.

Eqs. (C.42)-(C.43) express the preintegrated noise $\boldsymbol{\eta}_{ij}^{\Delta}$ as a linear function of the IMU measurement noise $\boldsymbol{\eta}_k^d \doteq [\boldsymbol{\eta}_k^{gd}, \boldsymbol{\eta}_k^{ad}]$, $k = 1, \ldots, j-1$. Therefore, from the knowledge of the covariance of $\boldsymbol{\eta}_k^d$ (given in the IMU specifications), we can compute the covariance of $\boldsymbol{\eta}_{ij}^{\Delta}$, namely $\boldsymbol{\Sigma}_{ij}$, by a simple linear propagation.

In Appendix C.10.1, we provide a more clever way to compute $\boldsymbol{\Sigma}_{ij}$. In particular, we show that $\boldsymbol{\Sigma}_{ij}$ can be conveniently computed in iterative form: as a new IMU measurement arrive we only update $\boldsymbol{\Sigma}_{ij}$, rather than recomputing it from scratch. The iterative computation leads to simpler expressions and is more amenable for online inference.

Incorporating Bias Updates

In the previous section, we assumed that the bias $\{\bar{\mathbf{b}}_i^a, \bar{\mathbf{b}}_i^g\}$ that is used during preintegration between $k = i$ and $k = j$ is correct and does not change. However, more likely, the bias estimate changes by a small amount $\delta\mathbf{b}$ during optimization. One solution would be to recompute the delta measurements when the bias changes; however, that is computationally expensive. Instead, given a bias update $\mathbf{b} \leftarrow \bar{\mathbf{b}} + \delta\mathbf{b}$, we can update the delta measurements using a first-order expansion:

$$\begin{aligned}
\Delta\tilde{\mathrm{R}}_{ij}(\mathbf{b}_i^g) &\simeq \Delta\tilde{\mathrm{R}}_{ij}(\bar{\mathbf{b}}_i^g)\,\mathrm{Exp}\left(\tfrac{\partial\Delta\bar{\mathrm{R}}_{ij}}{\partial\mathbf{b}^g}\delta\mathbf{b}^g\right) \\
\Delta\tilde{\mathbf{v}}_{ij}(\mathbf{b}_i^g, \mathbf{b}_i^a) &\simeq \Delta\tilde{\mathbf{v}}_{ij}(\bar{\mathbf{b}}_i^g, \bar{\mathbf{b}}_i^a) + \tfrac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^g}\delta\mathbf{b}_i^g + \tfrac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^a}\delta\mathbf{b}_i^a \\
\Delta\tilde{\mathbf{p}}_{ij}(\mathbf{b}_i^g, \mathbf{b}_i^a) &\simeq \Delta\tilde{\mathbf{p}}_{ij}(\bar{\mathbf{b}}_i^g, \bar{\mathbf{b}}_i^a) + \tfrac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}^g}\delta\mathbf{b}_i^g + \tfrac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}^a}\delta\mathbf{b}_i^a
\end{aligned} \tag{C.44}$$

This is similar to the bias correction in [Lupton and Sukkarieh, 2012] but operates directly on $\mathrm{SO}(3)$. The Jacobians $\{\frac{\partial\Delta\bar{\mathrm{R}}_{ij}}{\partial\mathbf{b}^g}, \frac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^g}, \ldots\}$ (computed at $\bar{\mathbf{b}}_i$, the bias estimate at integration time) describe how the measurements change due to a change in the bias estimate. The Jacobians remain constant and can be precomputed during the preintegration. The derivation of the Jacobians is very similar to the one we used in Section C.6.1 to express the measurements as a large value *plus* a small perturbation and is given in Appendix C.10.2.

Preintegrated IMU Factors

Given the preintegrated measurement model in (C.38) and since measurement noise is zero-mean and Gaussian (with covariance $\boldsymbol{\Sigma}_{ij}$) up to first order (C.39), it is now easy to write the residual errors $\mathbf{r}_{\mathcal{I}_{ij}} \doteq [\mathbf{r}_{\Delta\mathrm{R}_{ij}}^\mathsf{T}, \mathbf{r}_{\Delta\mathbf{v}_{ij}}^\mathsf{T}, \mathbf{r}_{\Delta\mathbf{p}_{ij}}^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^9$, where

$$\begin{aligned}
\mathbf{r}_{\Delta\mathrm{R}_{ij}} &\doteq \mathrm{Log}\left(\left(\Delta\tilde{\mathrm{R}}_{ij}(\bar{\mathbf{b}}_i^g)\mathrm{Exp}\left(\tfrac{\partial\Delta\bar{\mathrm{R}}_{ij}}{\partial\mathbf{b}^g}\delta\mathbf{b}^g\right)\right)^\mathsf{T}\mathrm{R}_i^\mathsf{T}\mathrm{R}_j\right) \\
\mathbf{r}_{\Delta\mathbf{v}_{ij}} &\doteq \mathrm{R}_i^\mathsf{T}\left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right) - \left[\Delta\tilde{\mathbf{v}}_{ij}(\bar{\mathbf{b}}_i^g, \bar{\mathbf{b}}_i^a) + \tfrac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^g}\delta\mathbf{b}^g + \tfrac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^a}\delta\mathbf{b}^a\right] \\
\mathbf{r}_{\Delta\mathbf{p}_{ij}} &\doteq \mathrm{R}_i^\mathsf{T}\left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \tfrac{1}{2}\mathbf{g}\Delta t_{ij}^2\right) - \left[\Delta\tilde{\mathbf{p}}_{ij}(\bar{\mathbf{b}}_i^g, \bar{\mathbf{b}}_i^a) + \tfrac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}^g}\delta\mathbf{b}^g + \tfrac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}_a}\delta\mathbf{b}^a\right],
\end{aligned} \tag{C.45}$$

in which we also included the bias updates of Eq. (C.44).

According to the "lift-solve-retract" method (Section C.3.3), at each Gauss-Newton iteration we need to re-parametrize (C.45) using the retraction (C.21). Then, the "solve" step requires to linearize the resulting cost around the current estimate. For the purpose of linearization, it is convenient to compute analytic expressions of the Jacobians of the residual errors, which we derive in the Appendix C.10.3.

Bias Model

When presenting the IMU model (C.27), we said that biases are slowly time-varying quantities. Hence, we model them with a "Brownian motion", *i.e.*, integrated white noise:

$$\dot{\mathbf{b}}^g(t) = \boldsymbol{\eta}^{bg}, \qquad \dot{\mathbf{b}}^a(t) = \boldsymbol{\eta}^{ba}. \tag{C.46}$$

Integrating (C.46) over the time interval $[t_i, t_j]$ between two consecutive keyframes i and j we get:

$$\mathbf{b}_j^g = \mathbf{b}_i^g + \boldsymbol{\eta}^{bgd}, \qquad \mathbf{b}_j^a = \mathbf{b}_i^a + \boldsymbol{\eta}^{bad}, \tag{C.47}$$

where, as done before, we use the shorthand $\mathbf{b}_i^g \doteq \mathbf{b}^g(t_i)$, and we define the discrete noises $\boldsymbol{\eta}^{bgd}$ and $\boldsymbol{\eta}^{bad}$, which have zero mean and covariance $\boldsymbol{\Sigma}^{bgd} \doteq \Delta t_{ij}\text{Cov}(\boldsymbol{\eta}^{bg})$ and $\boldsymbol{\Sigma}^{bad} \doteq \Delta t_{ij}\text{Cov}(\boldsymbol{\eta}^{ba})$, respectively (*cf.* [Crassidis, 2006, Appendix]).

The model (C.47) can be readily included in our factor graph, as a further additive term in (C.26) for all consecutive keyframes:

$$\|\mathbf{r}_{\mathbf{b}_{ij}}\|^2 \doteq \|\mathbf{b}_j^g - \mathbf{b}_i^g\|^2_{\boldsymbol{\Sigma}^{bgd}} + \|\mathbf{b}_j^a - \mathbf{b}_i^a\|^2_{\boldsymbol{\Sigma}^{bad}} \tag{C.48}$$

Structureless Vision Factors

In this section we introduce our structureless model for vision measurements. The key feature of our approach is the linear elimination of landmarks. Note that the elimination is repeated at each Gauss-Newton iteration, hence we are still guaranteed to obtain the optimal MAP estimate.

Visual measurements contribute to the cost (C.26) via the sum:

$$\sum_{i\in\mathcal{K}_k}\sum_{l\in\mathcal{C}_i}\|\mathbf{r}_{\mathcal{C}_{il}}\|^2_{\boldsymbol{\Sigma}_\mathcal{C}} = \sum_{l=1}^{L}\sum_{i\in\mathcal{X}(l)}\|\mathbf{r}_{\mathcal{C}_{il}}\|^2_{\boldsymbol{\Sigma}_\mathcal{C}} \tag{C.49}$$

which, on the right-hand-side, we rewrote as a sum of contributions of each landmark $l = 1, \ldots, L$. In (C.49), $\mathcal{X}(l)$ denotes the subset of keyframes in which l is seen.

A fairly standard model for the residual error of a single image measurement $\mathbf{z}_{il}$ is the reprojection error:

$$\mathbf{r}_{\mathcal{C}_{il}} = \mathbf{z}_{il} - \pi(\mathtt{R}_i, \mathbf{p}_i, \boldsymbol{æ}_l), \tag{C.50}$$

where $\boldsymbol{æ}_l \in \mathbb{R}^3$ denotes the position of the l-th landmark, and $\pi(\cdot)$ is a standard perspective projection, which also encodes the (known) IMU-camera transformation $\mathtt{T}_{\text{BC}}$.

Direct use of (C.50) would require to include the landmark positions $\boldsymbol{\text{æ}}_l$, $l = 1, \dots, L$ in the optimization, and this impacts negatively on computation. Therefore, in the following we adopt a *structureless* approach that avoids optimization over the landmarks, thus ensuring to retrieve the MAPestimate.

As recalled in Section C.3.3, at each GNiteration, we *lift* the cost function, using the retraction (C.21). For the vision factors this means that the original residuals (C.49) become:

$$\sum_{l=1}^{L} \sum_{i \in \mathcal{X}(l)} \| \mathbf{z}_{il} - \check{\pi}(\delta\boldsymbol{\phi}_i, \delta\mathbf{p}_i, \delta\boldsymbol{\text{æ}}_l) \|^2_{\boldsymbol{\Sigma}_\mathcal{C}} \tag{C.51}$$

where $\delta\boldsymbol{\phi}_i, \delta\mathbf{p}_i, \delta\boldsymbol{\text{æ}}_l$ are now Euclidean corrections, and $\check{\pi}(\cdot)$ is the lifted cost function. The "solve" step in the GNmethod is based on linearization of the residuals:

$$\sum_{l=1}^{L} \sum_{i \in \mathcal{X}(l)} \| \mathbf{F}_{il} \delta\mathbf{T}_i + \mathbf{E}_{il} \delta\boldsymbol{\text{æ}}_l - \mathbf{b}_{il} \|^2, \tag{C.52}$$

where $\delta\mathbf{T}_i \doteq [\delta\boldsymbol{\phi}_i \ \delta\mathbf{p}_i]^\mathsf{T}$; the Jacobians $\mathbf{F}_{il}, \mathbf{E}_{il}$, and the vector $\mathbf{b}_{il}$ (both normalized by $\boldsymbol{\Sigma}_\mathcal{C}^{1/2}$) result from the linearization. The vector $\mathbf{b}_{il}$ is the residual error at the linearization point.

Writing the second sum in (C.52) in matrix form we get:

$$\sum_{l=1}^{L} \| \mathbf{F}_l \ \delta\mathbf{T}_{\mathcal{X}(l)} + \mathbf{E}_l \ \delta\boldsymbol{\text{æ}}_l - \mathbf{b}_l \|^2 \tag{C.53}$$

where $\mathbf{F}_l, \mathbf{E}_l, \mathbf{b}_l$ are obtained by stacking $\mathbf{F}_{il}, \mathbf{E}_{il}, \mathbf{b}_{il}$, respectively, for all $i \in \mathcal{X}(l)$.

Since a landmark l appears in a single term of the sum (C.53), for any given choice of the pose perturbation $\delta\mathbf{T}_{\mathcal{X}(l)}$, the landmark perturbation $\delta\boldsymbol{\text{æ}}_l$ that minimizes the quadratic cost $\| \mathbf{F}_l \ \delta\mathbf{T}_{\mathcal{X}(l)} + \mathbf{E}_l \ \delta\boldsymbol{\text{æ}}_l - \mathbf{b}_l \|^2$ is:

$$\delta\boldsymbol{\text{æ}}_l = -(\mathbf{E}_l^\mathsf{T}\mathbf{E}_l)^{-1}\mathbf{E}_l^\mathsf{T}(\mathbf{F}_l \ \delta\mathbf{T}_{\mathcal{X}(l)} - \mathbf{b}_l) \tag{C.54}$$

Substituting (C.54) back into (C.53) we can *eliminate* the variable $\delta\boldsymbol{\text{æ}}_l$ from the optimization problem:

$$\sum_{l=1}^{L} \| (\mathbf{I} - \mathbf{E}_l(\mathbf{E}_l^\mathsf{T}\mathbf{E}_l)^{-1}\mathbf{E}_l^\mathsf{T}) \left(\mathbf{F}_l \ \delta\mathbf{T}_{\mathcal{X}(l)} - \mathbf{b}_l \right) \|^2, \tag{C.55}$$

where $\mathbf{I} - \mathbf{E}_l(\mathbf{E}_l^\mathsf{T}\mathbf{E}_l)^{-1}\mathbf{E}_l^\mathsf{T}$ is an orthogonal projector of $\mathbf{E}_l$. In Appendix C.10.4 we show that the cost (C.55) can be further manipulated, leading to a more efficient implementation.

This approach is well known in the bundle adjustment literature as the *Schur complement trick*, where a standard practice is to update the linearization point of $\boldsymbol{\text{æ}}_l$ via *back-substitution* [Hartley and Zisserman, 2004]. In contrast, we obtain the updated landmark

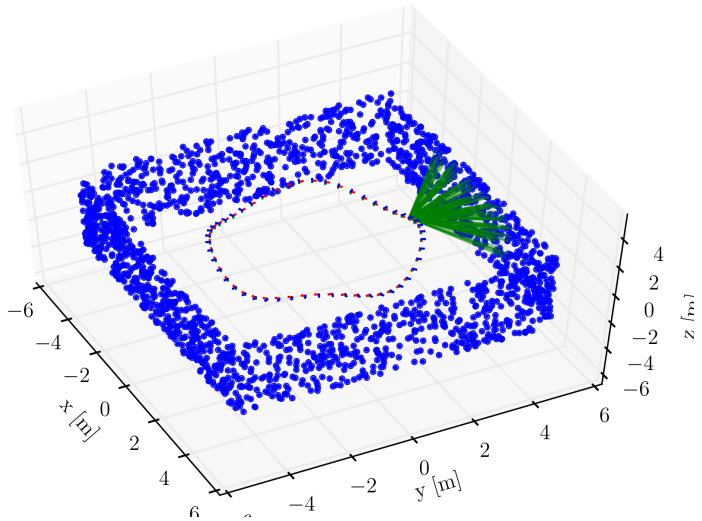


Figure C.5 – Simulation setup: The camera moves along a circular trajectory while observing features (green lines) on the walls of a square environment.

positions from the linearization point of the poses using a fast linear triangulation. Using this approach, we reduced a large set of factors (C.51) which involve poses and landmarks into a smaller set of L factors (C.55), which only involve poses. In particular, the factor corresponding to landmark l only involves the states $\mathcal{X}(l)$ observing l, creating the connectivity pattern of Fig. C.3. The same approach is also used in MSC-KF [Mourikis and Roumeliotis, 2007] to avoid the inclusion of landmarks in the state vector. However, since MSC-KF can only linearize and absorb a measurement once, the processing of measurements needs to be delayed until all measurements of the same landmark are observed. This does not apply to the proposed optimization-based approach, which allows for multiple relinearizations and the incremental inclusion of new measurements.

Experimental Analysis

We tested the proposed approach on both simulated and real data. Section C.8.1 reports simulation results, showing that our approach is accurate, fast, and consistent. Section C.8.2 compares our approach against the state-of-the-art, confirming its superior accuracy in real indoor and outdoor experiments.

Simulation Experiments

We simulated a camera following a circular trajectory of three meter radius with a sinusoidal vertical motion. The total length of the trajectory is 120 meters. While moving, the camera observes landmarks as depicted in Fig. C.5. The number of

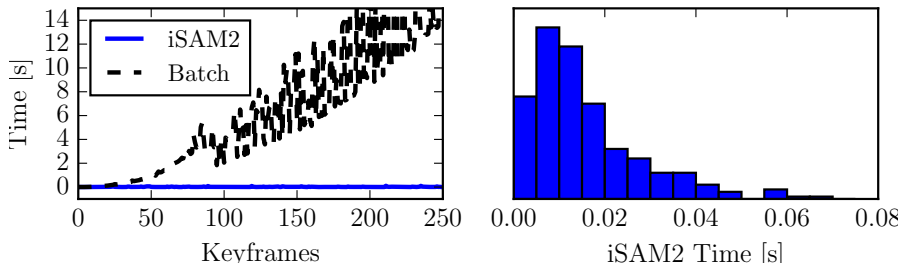


Figure C.6 – Left: CPU time required for inference, comparing batch estimation against iSAM2. Right: histogram plot of CPU time for the proposed approach.

landmark observations per frame is limited to 50. To simulate a realistic feature-tracker, we corrupt the landmark measurements with isotropic Gaussian noise with standard deviation $\sigma_{\text{px}} = 1$ pixel. The camera has a focal length of 315 pixels and runs at a rate of 2.5 Hz (simulating keyframes). The simulated acceleration and gyroscope measurements are computed from the analytic derivatives of the parametric trajectory and additionally corrupted by white noise and a slowly time-varying bias terms, according to the IMU model in Eq. (C.27).[2] To evaluate our approach, we performed a Monte Carlo analysis with 50 simulation runs, each with different realizations of process and measurement noise. In each run we compute the MAP estimate using the IMU and the vision models presented in this paper. The optimization (whose solution is the MAP estimate) is solved using the incremental smoothing algorithm iSAM2 [Kaess et al., 2012]. iSAM2 uses the Bayes tree [Kaess et al., 2010] data structure to obtain efficient variable ordering that minimizes fill-in in the square-root information matrix and, thus, minimizes computation time. Further, iSAM2 exploits the fact that new measurements often have only local effect on the MAP estimate, hence applies incremental updates directly to the square-root information matrix, only re-solving for the variables affected by a new measurement.

In the following we present the results of our experiments, organized in four subsections: 1) pose estimation accuracy and timing, 2) consistency, 3) bias estimation accuracy, and 4) first-order bias correction. Then, in Section C.8.1 we compare our approach against the original proposal of [Lupton and Sukkarieh, 2012].

Pose Estimation Accuracy and Timing

The optimal MAP estimate is given by the *batch* nonlinear optimization of the least-squares objective in Eq. (C.26). However, as shown on the left in Fig. C.6, the computational cost of batch optimization quickly increases as the trajectory length grows. A key

[2] We used the following IMU parameters: Gyroscope and accelerometer continuous-time noise density: $\sigma^g = 0.0007$ $[\text{rad}/(\text{s}\sqrt{\text{Hz}})]$, $\sigma^a = 0.019$ $[\text{m}/(\text{s}^2\sqrt{\text{Hz}})]$. Gyroscope and accelerometer *bias* continous-time noise density: $\sigma^{bg} = 0.0004$ $[\text{rad}/(\text{s}^2\sqrt{\text{Hz}})]$, $\sigma^{ba} = 0.012$ $[\text{m}/(\text{s}^3\sqrt{\text{Hz}})]$.

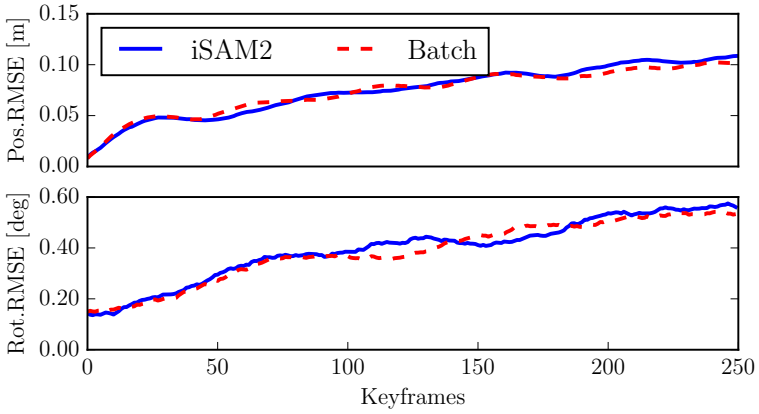


Figure C.7 – Root Mean Squared Error (RMSE) averaged over 50 Monte Carlo experiments, comparing batch nonlinear optimization and iSAM2.

ingredient that makes our approach extremely efficient is the use of the incremental smoothing algorithm iSAM2 [Kaess et al., 2012], which performs close-to-optimal inference, while preserving real-time capability. Fig. C.7 shows that the accuracy of iSAM2 is practically the same as the batch estimate. In odometry problems, the iSAM2 algorithm results in approximately constant update time per frame (Fig. C.6, left), which in our experiment is approximately 10 milliseconds per update (Fig. C.6, right).

Consistency

For generic motion, the VINproblem has four unobservable degrees of freedom, three corresponding to the global translation and one to the global orientation around the gravity direction (yaw), see [Kottas et al., 2012]. A VINalgorithm must preserve these observability properties and avoid inclusion of spurious information along the unobservable directions, which would result in inconsistency [Kottas et al., 2012]. Fig. C.8 reports orientation and position errors with the corresponding 3σ bounds, confirming that our approach is consistent. In the VINproblem, the gravity direction is observable, hence the uncertainty on roll and pitch remains bounded. In contrast, global yaw and position cannot be measured and the uncertainty slowly grows over time.

To present more substantial evidence of the fact that our estimator is consistent, we recall a standard measure of consistency, the *average* Normalized Estimation Error Squared (NEES) [Bar-Shalom et al., 2001]. The NEES is the squared estimation error $\boldsymbol{\epsilon}_k$ normalized by the estimator-calculated covariance $\boldsymbol{\Sigma}_k$:

$$\eta_k \doteq \boldsymbol{\epsilon}_k^{\mathsf{T}} \hat{\boldsymbol{\Sigma}}_k^{-1} \boldsymbol{\epsilon}_k \qquad \text{(NEES)} \tag{C.56}$$

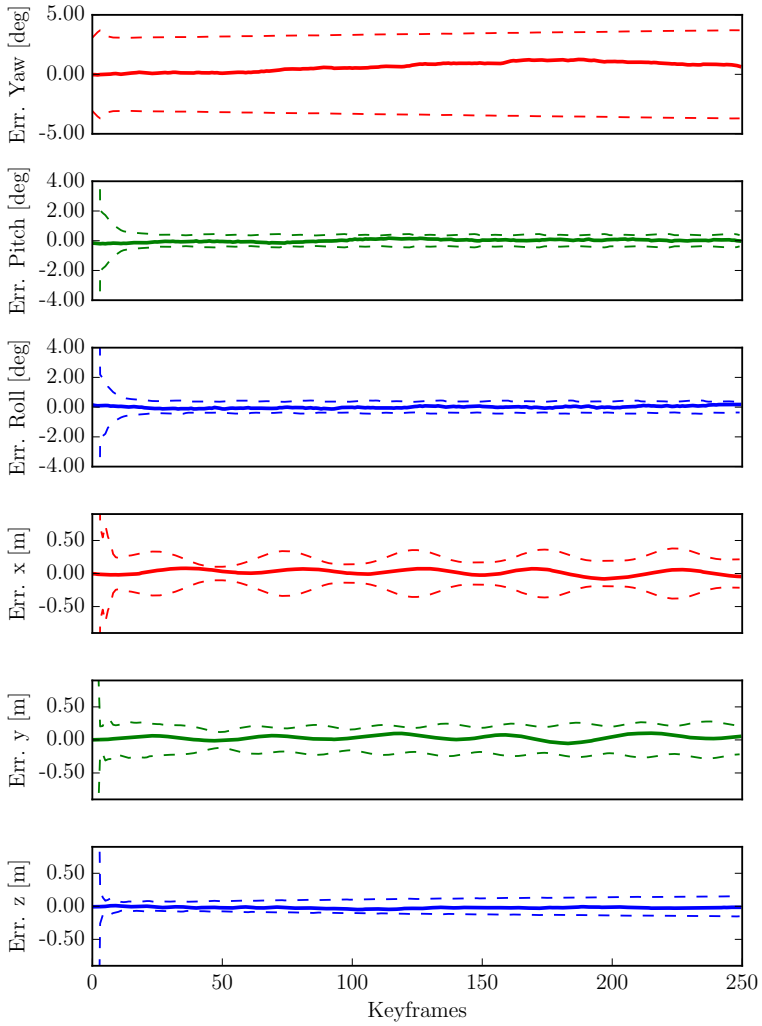


Figure C.8 – Orientation and position errors with 3σ bounds (single simulation).

The error in estimating the current pose is computed as:

$$\boldsymbol{\epsilon}_k \doteq \left[\mathrm{Log}(\hat{\mathrm{R}}_k^{\mathsf{T}}\mathrm{R}_k^{\mathrm{gt}}),\ \hat{\mathrm{R}}_k^{\mathsf{T}}(\hat{\mathbf{p}}_k - \mathbf{p}_k^{\mathrm{gt}})\right]^{\mathsf{T}} \tag{C.57}$$

where the exponent "gt" denotes ground-truth states and $(\hat{\mathrm{R}}_k, \hat{\mathbf{p}}_k)$ denotes the estimated pose at time k. Note that the error (C.57) is expressed in the body frame and it is consistent with our choice of the retraction in Eq. (C.21) (intuitively, the retraction applies the perturbation in the body frame).

The *average* NEES over N independent Monte Carlo runs, can be computed by averaging the NEES values:

$$\bar{\eta}_k = \tfrac{1}{N}\textstyle\sum_{i=1}^{N} \eta_k^{(i)} \qquad \text{(average NEES)} \tag{C.58}$$

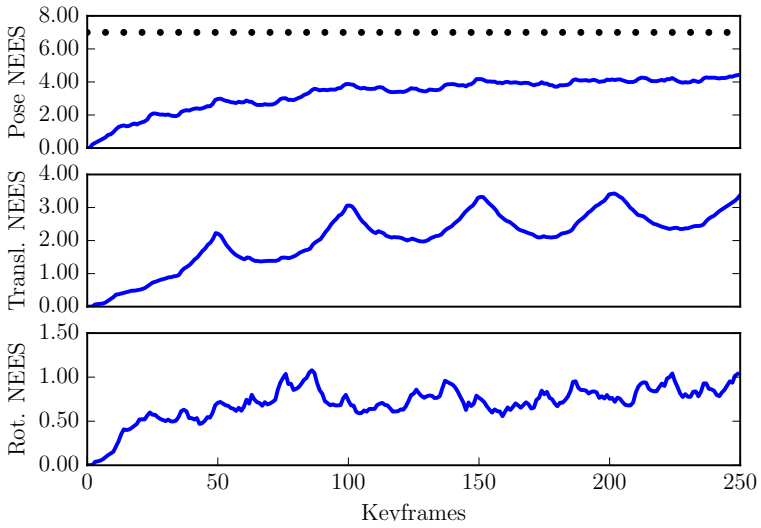


Figure C.9 – Normalized Estimation Error Squared (NEES) averaged over 50 Monte Carlo runs. The average NEES is reported for the current pose (top), current position (middle), and current rotation (bottom).

where $\eta_k^{(i)}$ is the NEES computed at the i-th Monte Carlo run. If the estimator is consistent, then $N\bar{\eta}_k$ is χ^2_n chi-square distributed with $n = \dim(\boldsymbol{\epsilon}_k) \cdot N$ degrees of freedom [Bar-Shalom et al., 2001, pp. 234]. We evaluate this hypothesis with a χ^2_n acceptance test [Bar-Shalom et al., 2001, pp. 235]. For a significance level $\alpha = 2.5\%$ and $n = \dim(\boldsymbol{\epsilon}_k) \cdot N = 6 \cdot 50$, the acceptance region of the test is given by the two-sided probability concentration region $\bar{\eta}_k \in [5.0, 7.0]$. If $\bar{\eta}_k$ rises significantly higher than the upper bound, the estimator is overconfident, if it tends below the lower bound, it is conservative. In VINone usually wants to avoid overconfident estimators: the fact that $\bar{\eta}_k$ exceeds the upper bound is an indicator of the fact that the estimator is including spurious information in the inference process.

In Fig. C.9 we report the average NEES of the proposed approach. The average NEES approaches the lower bound but, more importantly, it remains below the upper bound at 7.0 (black dots), which assures that the estimator is not overconfident. We also report the average rotational and translational NEES to allow a comparison with the observability-constrained EKF in [Kottas et al., 2012, Hesch et al., 2014], which obtains similar results by enforcing explicitly the observability properties in EKF.

Bias Estimation Accuracy

Our simulations allow us to compare the estimated gyroscope and accelerometer bias with the true biases that were used to corrupt the simulated inertial measurements. Fig. C.10 shows that biases estimated by our approach (in blue) correctly track the ground truth biases (in red). Note that, since we use a smoothing approach, at each

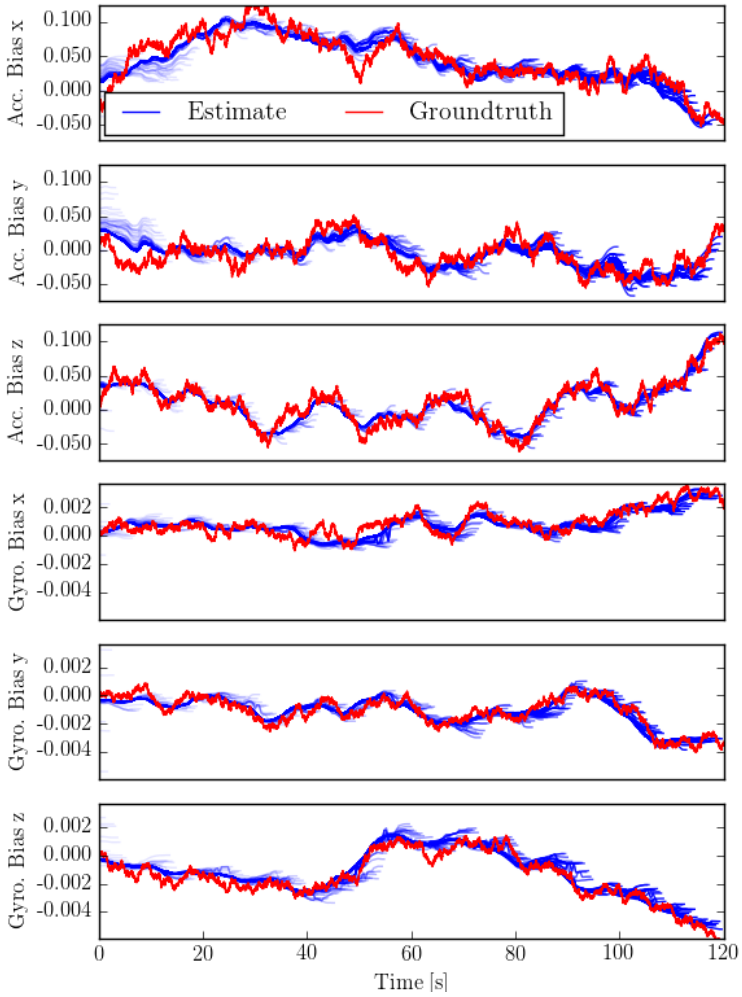


Figure C.10 – Comparison between ground truth bias (red line) and estimated bias (blue lines) in a Monte Carlo run.

step, we potentially change the entire history of the bias estimates, hence we visualize the bias estimates using multiple curves. Each curve represents the history of the estimated biases from time zero (left-most extreme of the blue curve) to the current time (right-most extreme of the blue curve).

First-Order Bias Correction

We performed an additional Monte-Carlo analysis to evaluate the a-posteriori bias correction proposed in Section C.6.3. The preintegrated measurements are computed with the bias estimate at the time of integration. However, as seen in Fig. C.10, the bias estimate for an older preintegrated measurement may change when more information becomes available. To avoid repeating the integration when the bias estimate changes,

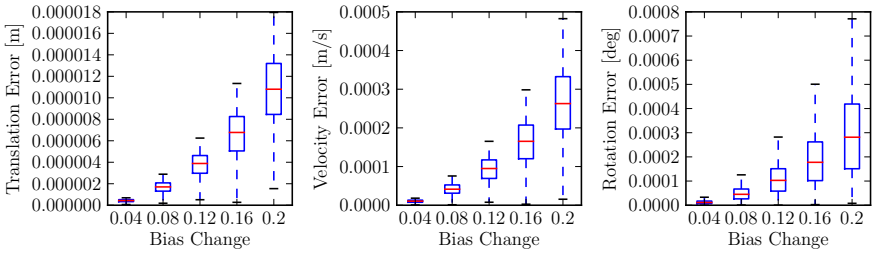


Figure C.11 – Error committed when using the first-order approximation (C.44) instead of repeating the integration, for different bias perturbations. Left: $\Delta\tilde{\mathbf{p}}_{ij}(\bar{\mathbf{b}}_i + \delta\mathbf{b}_i)$ error; Center: $\Delta\tilde{\mathbf{v}}_{ij}(\bar{\mathbf{b}}_i + \delta\mathbf{b}_i)$ error; Right: $\Delta\tilde{\mathrm{R}}_{ij}(\bar{\mathbf{b}}_i + \delta\mathbf{b}_i)$ error. Statistics are computed over 1000 Monte Carlo runs.

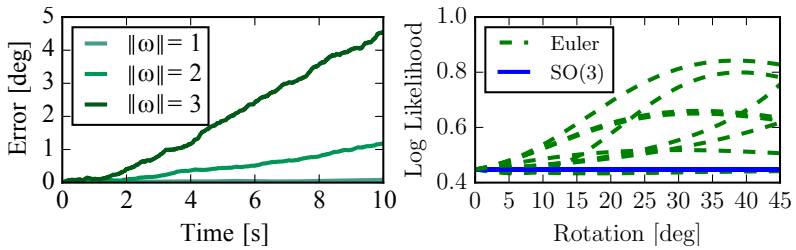
we perform a first-order correction of the preintegrated measurement according to Eq. (C.44). The accuracy of this first order bias correction is reported in Fig. C.11. To compute the statistics, we integrated 100 random IMU measurements with a given bias estimate $\bar{\mathbf{b}}_i$ which results in the preintegrated measurements $\Delta\tilde{\mathrm{R}}_{ij}(\bar{\mathbf{b}}_i), \Delta\tilde{\mathbf{v}}_{ij}(\bar{\mathbf{b}}_i)$ and $\Delta\tilde{\mathbf{p}}_{ij}(\bar{\mathbf{b}}_i)$. Subsequently, a random perturbation $\delta\mathbf{b}_i$ with magnitude between 0.04 and 0.2 was applied to both the gyroscope and accelerometer bias. We repeated the integration at $\bar{\mathbf{b}}_i + \delta\mathbf{b}_i$ to obtain $\Delta\tilde{\mathrm{R}}_{ij}(\bar{\mathbf{b}}_i + \delta\mathbf{b}_i), \Delta\tilde{\mathbf{v}}_{ij}(\bar{\mathbf{b}}_i + \delta\mathbf{b}_i)$ and $\Delta\tilde{\mathbf{p}}_{ij}(\bar{\mathbf{b}}_i + \delta\mathbf{b}_i)$. This ground-truth result was then compared against the first-order correction in (C.44) to compute the error of the approximation. The errors resulting from the first-order approximation are negligible, even for relatively large bias perturbations.

Advantages over the Euler-angle-based formulation

In this section we compare the proposed IMU preintegration with the original formulation of [Lupton and Sukkarieh, 2012], based on Euler angles. We observe three main problems with the preintegration using Euler angles, which are avoided in our formulation.

The first drawback is that, in contrast to the integration using the exponential map in Eq. (C.30), the rotation integration based on Euler angles is only exact up to the first order. For the interested reader, we recall the rotation rate integration using Euler angles in Appendix C.10.5. On the left of Fig. C.12, we report the integration errors commited by the Euler angle parametrization when integrating angular rates with randomly selected rotation axes and magnitude in the range from 1 to 3 rad/s. Integration error in Euler angles accumulates quickly when the sampling time Δt or the angular rate $\tilde{\boldsymbol{\omega}}$ are large. On the other hand, the proposed approach, which performs integration directly on the rotation manifold, is exact, regardless the values of Δt and $\tilde{\boldsymbol{\omega}}$.

The second drawback is that the Euler parametrization is not *fair* [Hornegger, 1997], which means that, given the preintegrated Euler angles $\tilde{\boldsymbol{\theta}}$, the negative log-likelihood



(a) (b)

Figure C.12 – (a) Integration errors committed with the Euler angle parametrization for angular velocities ω of increasing magnitude [rad/s]. (b) Negative log-likelihood of a rotation measurement under the action of random rigid body transformations.

$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2}\|\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}\|^2_\Sigma$ is not invariant under the action of rigid body transformations. On the right of Fig. C.12 we show experimentally how the log-likelihood changes when the frame of reference is rotated around randomly selected rotation axes. This essentially means that an estimator using Euler angles may give different results for different choices of the world frame (*cf.* with Fig. C.2). On the other hand, the $\mathrm{SO}(3)$ parametrization can be easily seen to be fair (the negative likelihood (C.16) can be promptly seen to be left invariant), and this is confirmed by Fig. C.12 (right).

The third drawback is the existence of so-called *gimball lock* singularities. For a zyx Euler angle parametrization, the singularity is reached at pitch values of $\theta = \frac{\pi}{2} + n\pi$, for $n \in \mathbb{Z}$. To evaluate the effect of the singularity and how it affects the computation of preintegrated measurement noise, we performed the following Monte Carlo analysis. We simulated a set of trajectories that reach maximum pitch values $\theta_{\max}$ of increasing magnitude. For each trajectory, we integrate the rotation uncertainty using the Euler parametrization and the proposed on-manifold approach. The ground-truth covariance is instead obtained through sampling. We use the Kullback-Leibler (KL) divergence to quantify the mismatch between the estimated covariances and the ground-truth one. The results of this experiment are shown in Fig. C.13, where we observe that the closer we get to the singularity, the worse is the noise propagation using Euler angles. On the other hand, the proposed approach can accurately estimate the measurement covariance, independently on the motion of the platform.

Real Experiments

We integrated the proposed inertial factors in a monocular VINpipeline to benchmark its performance against the state of the art. In the following, we first discuss our implementation, and then present results from an indoor experiment with motion-capture ground-truth. Finally, we show results from longer trajectories in outdoor experiments. The results confirm that our approach is more accurate than state-of-the-art filtering and fixed-lag smoothing algorithms, and enables fast inference in

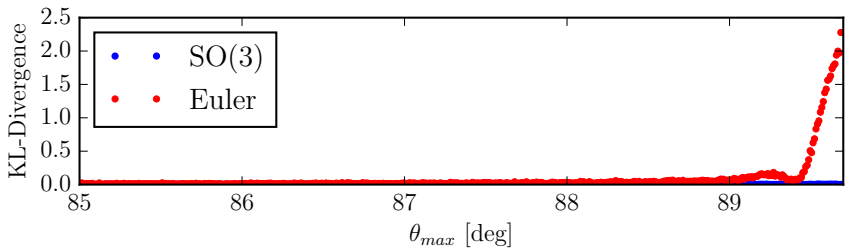


Figure C.13 – Kullback-Leibler divergence between the preintegrated rotation covariance – computed using Euler angles (red) and the proposed approach (blue)– and the ground-truth covariance. The Euler angle parametrization degrades close to the singularity at $\theta_{\max} = 90$ deg while the proposed on-manifold approach is accurate regardless of the motion.

real-world problems.

Implementation

Our implementation consists of a high frame rate tracking front-end based on SVO[3] Forster et al. [2014b] and an optimization back-end based on iSAM2 Kaess et al. [2012][4]. The front-end tracks salient features in the image at camera rate while the back-end optimizes in parallel the state of selected *keyframes* as described in this paper.

SVO Forster et al. [2014b] is a precise and robust monocular visual odometry system that employs *sparse image alignment* to estimate incremental motion and tracks features by minimizing the photometric error between subsequent frames. The difference to tracking features individually, as in standard Lucas-Kanade tracking, is that we exploit the known depth of features from previous triangulations. This allows us to track all features as a bundle in a single optimization that satisfies epipolar constraints; hence, outliers only originate from erroneous triangulations. In the visual-inertial setting, we further exploit the availability of accurate rotation increments, obtained by integrating angular velocity measurements from the gyroscope. These increments are used as rotation priors in the sparse-image-alignment algorithm, and this increases the overall robustness of the system. The motion estimation is combined with an outlier resistant probabilistic triangulation method that is implemented with a recursive Bayesian filter. The high frame-rate motion estimation combined with the robust depth estimation results in increased robustness in scenes with repetitive and high frequency texture (*e.g.*, asphalt). The output of SVO are selected keyframes with feature-tracks corresponding to triangulated landmarks. This data is passed to the back-end that computes the visual-inertial MAP estimate in Eq. (C.26) using iSAM2 Kaess et al. [2012].

We remark that our approach does not marginalize out past states. Therefore, while the approach is designed for fast visual-inertial odometry, if desired, it could be readily

[3] http://github.com/uzh-rpg/rpg_svo
[4] http://borg.cc.gatech.edu

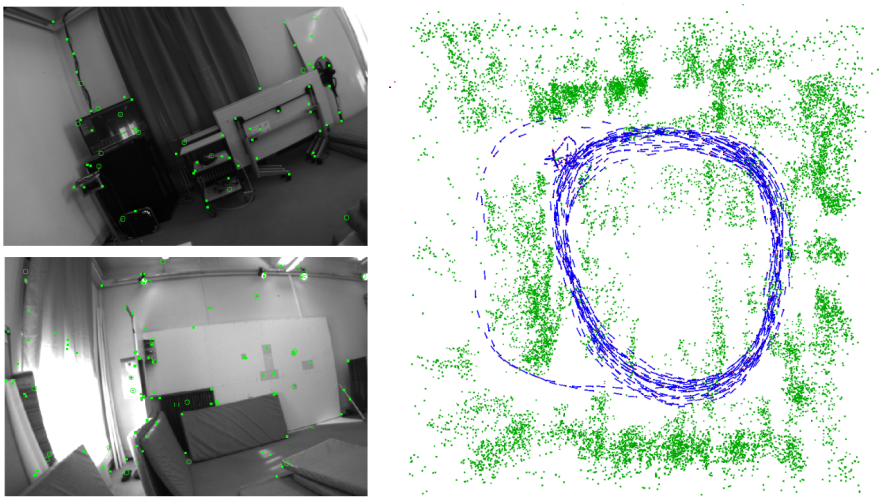


Figure C.14 – Left: two images from the indoor trajectory dataset with tracked features in green. Right: top view of the trajectory estimate produced by our approach (blue) and 3D landmarks triangulated from the trajectory (green).

extended to incorporate loop closures.

Indoor Experiments

The indoor experiment shows that the proposed approach is more accurate than two competitive state-of-the-art approaches, namely OKVIS[5] [Leutenegger et al., 2015], and MSCKF [Mourikis and Roumeliotis, 2007]. The experiment is performed on the 430m-long indoor trajectory of Fig. C.14. The dataset was recorded with a forward-looking VI-Sensor [Nikolic et al., 2014] that consists of an ADIS16448 MEMS IMU and two embedded WVGA monochrome cameras (we only use the left camera). Intrinsic and extrinsic calibration was obtained using [Furgale et al., 2013]. The camera runs at 20Hz and the IMU at 800Hz. Ground truth poses are provided by a Vicon system mounted in the room; the *hand-eye* calibration between the Vicon markers and the camera is computed using a least-squares method [Park and Martin, 1994].

Fig. C.15 compares the proposed system against the OKVIS algorithm [Leutenegger et al., 2015], and an implementation of the MSCKF filter [Mourikis and Roumeliotis, 2007]. Both these algorithms currently represent the state-of-the-art in VIN, OKVIS for optimization-based approaches, and MSCKF for filtering methods. We obtained the datasets as well as the trajectories computed with OKVIS and MSCKF from the authors of [Leutenegger et al., 2015]. We use the relative error metrics proposed in [Geiger et al., 2012] to obtain error statistics. The metric evaluates the relative error by averaging the drift over trajectory segments of different length ($\{10, 40, 90, 160, 250, 360\}$m in

[5] https://github.com/ethz-asl/okvis

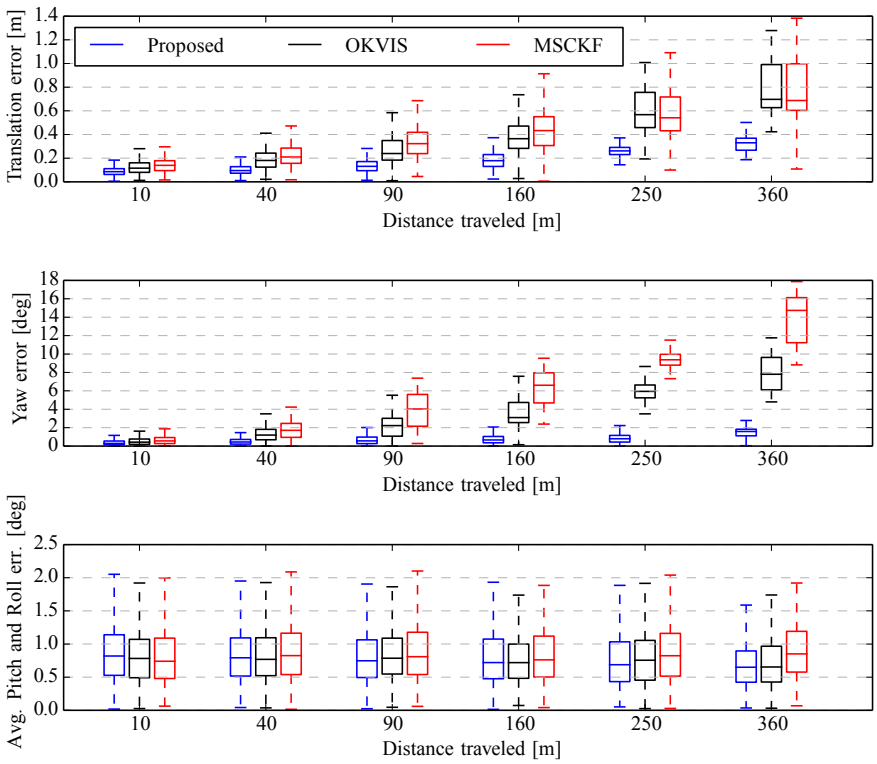


Figure C.15 – Comparison of the proposed approach versus the OKVIS algorithm [Leutenegger et al., 2015] and an implementation of the MSCKF filter [Mourikis and Roumeliotis, 2007]. Relative errors are measured over different segments of the trajectory, of length $\{10, 40, 90, 160, 250, 360\}$m, according to the odometric error metric in [Geiger et al., 2012].

Fig. C.15). Our approach exhibits less drift than the state-of-the-art, achieving 0.3m drift on average over 360m traveled distance; OKVIS and MSCKF accumulate an average error of 0.7m. We observe significantly less drift in yaw direction in the proposed approach while the error in pitch and and roll direction is constant for all methods due to the observability of the gravity direction.

We highlight that these three algorithms use different front-end feature tracking systems, which influence the overall performance of the approach. Therefore, while in Section C.8.1 we discussed only aspects related to the preintegration theory, in this section we evaluate the proposed system as a whole (SVO, preintegration, structureless vision factors, iSAM2).

Evaluating consistency in real experiments by means of analysing the average NEES is difficult as one would have to evalate and average the results of multiple runs of the same trajectory with different realizations of sensor noise. In Figure C.16 we show the error plots with the 3-sigma bounds for a single run. The result is consistent as the estimation errors remain within the bounds of the estimated uncertainty. In this

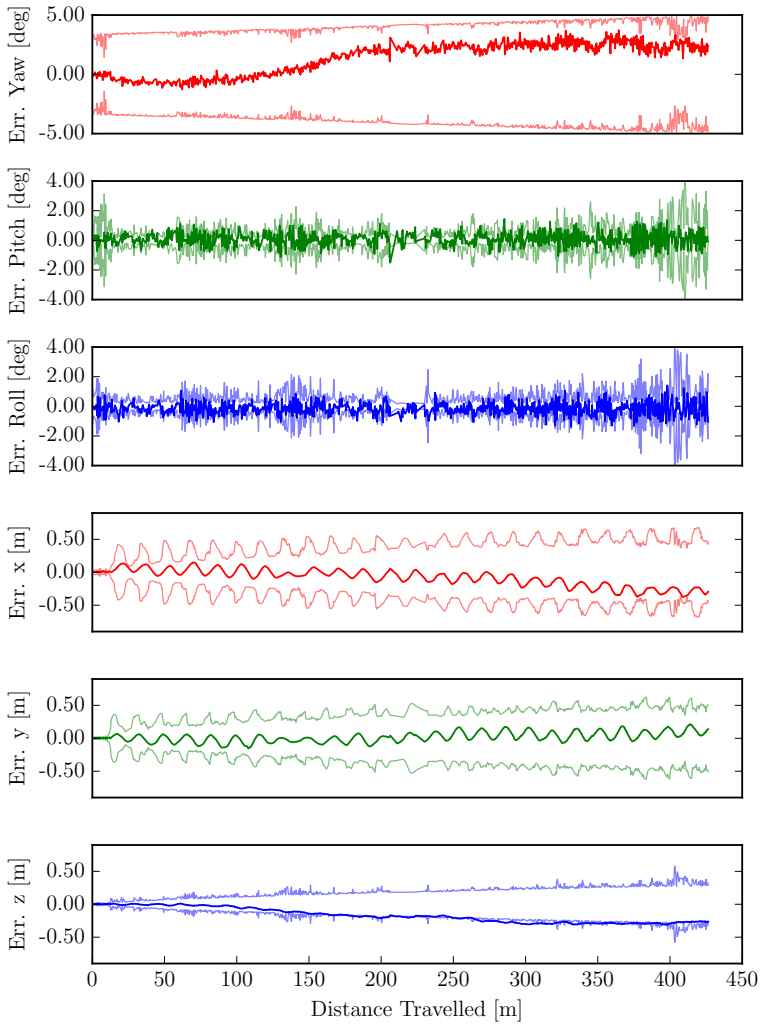


Figure C.16 – Orientation and position errors with 3σ bounds for the real indoor experiment in Fig. C.14.

experiment, we aligned only the first frame of the trajectory with the vicon trajectory. Therefore, analyzing the drift over 400 meters is very prone to errors in the initial pose from the ground-truth or errors in the hand-eye calibration of the system.

Figure C.17 illustrates the time required by the back-end to compute the full MAPestimate, by running iSAM2 with 10 optimization iterations. The experiment was performed on a standard laptop (Intel i7, 2.4 GHz). The average update time for iSAM2 is 10ms. The peak corresponds to the start of the experiment in which the camera was not moving. In this case the number of tracked features becomes very large making the back-end slightly slower. The SVO front-end requires approximately 3ms to process a frame on the laptop while the back-end runs in a parallel thread and optimizes only keyframes. Although the processing times of OKVIS were not reported, the approach

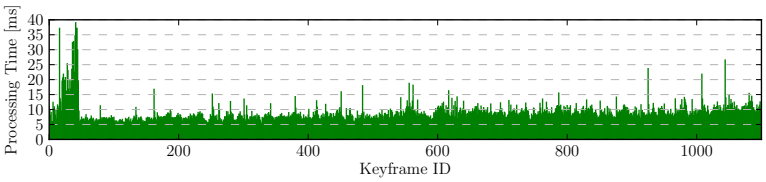


Figure C.17 – Processing-time per keyframe for the proposed VINapproach.

is described as computationally demanding [Leutenegger et al., 2015]. OKVIS needs to repeat IMU integration at every change of the linearization point, which we avoid by using the preintegrated IMU measurements.

Outdoor Experiments

The second experiment is performed on an outdoor trajectory, and compares the proposed approach against the *Google* Tango*Peanut* sensor (mapper version 3.15), which is an *engineered* VINsystem. We rigidly attached the VI-Sensor to a Tangodevice and walked around an office building. Fig. C.18 depicts the trajectory estimates for our approach and Google Tango. The trajectory starts and ends at the same location, hence we can report the end-to-end error which is 1.5m for the proposed approach and 2.2m for the Google Tangosensor.

In Fig. C.18 we also show the estimated landmark positions (in green). 3D points are not estimated by our approach (which uses a structureless vision model), but are triangulated from our trajectory estimate for visualization purposes.

The third experiment is the one in Fig. C.19. The trajectory goes across three floors of an office building and eventually returns to the initial location on the ground floor. Also in this case the proposed approach guarantees a very small end-to-end error (0.5m), while Tangoaccumulates 1.4m error.

We remark that Tangoand our system use different sensors, hence the reported end-to-end errors only allow for a qualitative comparison. However, the IMUs of both sensors exhibit similar noise characteristics [tan, adi] and the Tangocamera has a significantly larger field-of-view and better shutter speed control than our sensor. Therefore, the comparison is still valuable to assess the accuracy of the proposed approach.

A video demonstrating the execution of our approach for the real experiments discussed in this section can be viewed at https://youtu.be/CsJkci5lfco

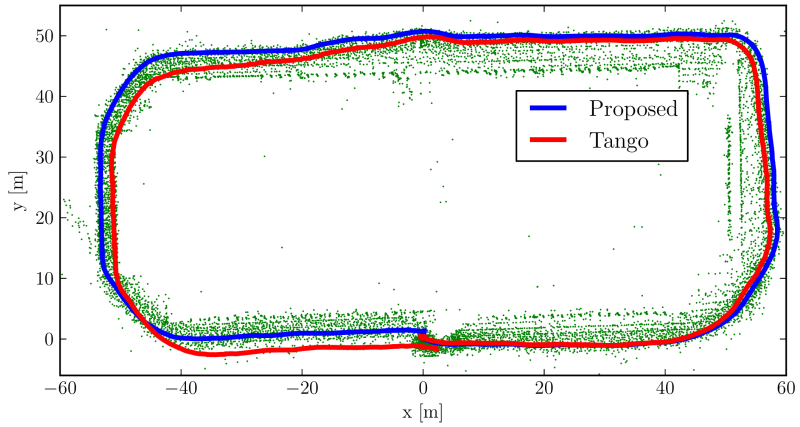


Figure C.18 – Outdoor trajectory (length: 300m) around a building with identical start and end point at coordinates $(0, 0, 0)$. The end-to-end error of the proposed approach is 1.0m. Google Tango accumulated 2.2m drift. The green dots are the 3D points triangulated from our trajectory estimate.

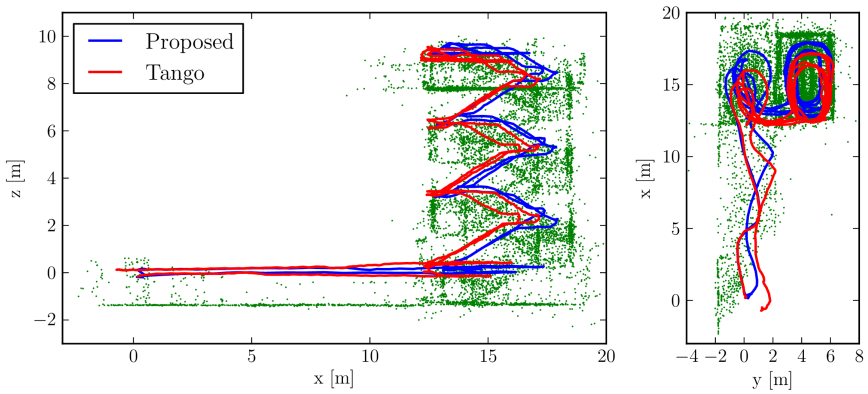


Figure C.19 – Real test comparing the proposed VINapproach against Google Tango. The 160m-long trajectory starts at $(0, 0, 0)$ (ground floor), goes up till the 3rd floor of a building, and returns to the initial point. The figure shows a side view (left) and a top view (right) of the trajectory estimates for our approach (blue) and Tango(red). Google Tangoaccumulates 1.4m error, while the proposed approach only has 0.5m drift. 3D points triangulated from our trajectory estimate are shown in green for visualization purposes.

Conclusion

This paper proposes a novel preintegration theory, which provides a grounded way to model a large number of IMU measurements as a single motion constraint. Our proposal improves over related works that perform integration in a global frame, e.g., [Leutenegger et al., 2013, Mourikis and Roumeliotis, 2007], as we do not commit to a linearization point during integration. Moreover, it leverages the seminal work on

preintegration [Lupton and Sukkarieh, 2012], bringing to maturity the preintegration and uncertainty propagation in SO(3).

As a second contribution, we discuss how to use the preintegrated IMU model in a VINpipeline; we adopt a structureless model for visual measurements which avoids optimizing over 3D landmarks. Our VINapproach uses iSAM2 to perform constant-time incremental smoothing.

An efficient implementation of our approach requires 10ms to perform inference (back-end), and 3ms for feature tracking (front-end). Experimental results also confirm that our approach is more accurate than state-of-the-art alternatives, including filtering and optimization-based techniques.

We release the source-code of the IMU preintegration and the structurless vision factors in the GTSAM 4.0 optimization toolbox [Dellaert, 2012] and provide additional theoretical derivations and implementation details in the Appendix of this paper.

Appendix

Iterative Noise Propagation

In this section we show that the computation of the preintegrated noise covariance, discussed in Section C.6.2, can be carried out in iterative form, which leads to simpler expressions and is more amenable for online inference.

Let us start from the preintegrated rotation noise in (C.42). To write $\delta\boldsymbol{\phi}_{ij}$ in iterative form, we simply take the last term ($k = j-1$) out of the sum and rearrange the terms:

$$
\begin{aligned}
\delta\boldsymbol{\phi}_{ij} &\simeq \sum_{k=i}^{j-1} \Delta\tilde{\mathrm{R}}_{k+1j}^{\mathsf{T}} \mathrm{J}_r^k \boldsymbol{\eta}_k^{gd} \Delta t \qquad \text{(C.59)} \\
&= \sum_{k=i}^{j-2} \Delta\tilde{\mathrm{R}}_{k+1j}^{\mathsf{T}} \mathrm{J}_r^k \boldsymbol{\eta}_k^{gd} \Delta t + \overbrace{\Delta\tilde{\mathrm{R}}_{jj}^{\mathsf{T}}}^{=\mathbf{I}_{3\times 3}} \mathrm{J}_r^{j-1} \boldsymbol{\eta}_{j-1}^{gd} \Delta t \\
&= \sum_{k=i}^{j-2} (\overbrace{\Delta\tilde{\mathrm{R}}_{k+1j-1}\Delta\tilde{\mathrm{R}}_{j-1j}}^{=\Delta\tilde{\mathrm{R}}_{k+1j}})^{\mathsf{T}} \mathrm{J}_r^k \boldsymbol{\eta}_k^{gd} \Delta t + \mathrm{J}_r^{j-1} \boldsymbol{\eta}_{j-1}^{gd} \Delta t \\
&= \Delta\tilde{\mathrm{R}}_{j-1j}^{\mathsf{T}} \sum_{k=i}^{j-2} \Delta\tilde{\mathrm{R}}_{k+1j-1}^{\mathsf{T}} \mathrm{J}_r^k \boldsymbol{\eta}_k^{gd} \Delta t + \mathrm{J}_r^{j-1} \boldsymbol{\eta}_{j-1}^{gd} \Delta t \\
&= \Delta\tilde{\mathrm{R}}_{j-1j}^{\mathsf{T}} \delta\boldsymbol{\phi}_{ij-1} + \mathrm{J}_r^{j-1} \boldsymbol{\eta}_{j-1}^{gd} \Delta t.
\end{aligned}
$$

Repeating the same process for $\delta\mathbf{v}_{ij}$ in (C.43):

$$\begin{aligned}\delta\mathbf{v}_{ij} &= \sum_{k=i}^{j-1}\left[-\Delta\tilde{\mathrm{R}}_{ik}\left(\tilde{\mathbf{a}}_k-\mathbf{b}_i^a\right)^\wedge\delta\boldsymbol{\phi}_{ik}\Delta t+\Delta\tilde{\mathrm{R}}_{ik}\boldsymbol{\eta}_k^{ad}\Delta t\right] \\ &= \sum_{k=i}^{j-2}\left[-\Delta\tilde{\mathrm{R}}_{ik}\left(\tilde{\mathbf{a}}_k-\mathbf{b}_i^a\right)^\wedge\delta\boldsymbol{\phi}_{ik}\Delta t+\Delta\tilde{\mathrm{R}}_{ik}\boldsymbol{\eta}_k^{ad}\Delta t\right] \\ &\quad -\Delta\tilde{\mathrm{R}}_{ij-1}\left(\tilde{\mathbf{a}}_{j-1}-\mathbf{b}_i^a\right)^\wedge\delta\boldsymbol{\phi}_{ij-1}\Delta t+\Delta\tilde{\mathrm{R}}_{ij-1}\boldsymbol{\eta}_{j-1}^{ad}\Delta t \\ &= \delta\mathbf{v}_{ij-1}-\Delta\tilde{\mathrm{R}}_{ij-1}\left(\tilde{\mathbf{a}}_{j-1}-\mathbf{b}_i^a\right)^\wedge\delta\boldsymbol{\phi}_{ij-1}\Delta t+\Delta\tilde{\mathrm{R}}_{ij-1}\boldsymbol{\eta}_{j-1}^{ad}\Delta t\end{aligned} \tag{C.60}$$

Doing the same for $\delta\mathbf{p}_{ij}$ in (C.43), and noting that $\delta\mathbf{p}_{ij}$ can be written as a function of $\delta\mathbf{v}_{ij}$ (*cf.* with the expression of $\delta\mathbf{v}_{ij}$ in (C.43)):

$$\begin{aligned}\delta\mathbf{p}_{ij} &= \sum_{k=i}^{j-1}\left[\delta\mathbf{v}_{ik}\Delta t-\frac{1}{2}\Delta\tilde{\mathrm{R}}_{ik}\left(\tilde{\mathbf{a}}_k-\mathbf{b}_i^a\right)^\wedge\delta\boldsymbol{\phi}_{ik}\Delta t^2+\frac{1}{2}\Delta\tilde{\mathrm{R}}_{ik}\boldsymbol{\eta}_k^{ad}\Delta t^2\right] \\ &= \sum_{k=i}^{j-2}\left[\delta\mathbf{v}_{ik}\Delta t-\frac{1}{2}\Delta\tilde{\mathrm{R}}_{ik}\left(\tilde{\mathbf{a}}_k-\mathbf{b}_i^a\right)^\wedge\delta\boldsymbol{\phi}_{ik}\Delta t^2+\frac{1}{2}\Delta\tilde{\mathrm{R}}_{ik}\boldsymbol{\eta}_k^{ad}\Delta t^2\right] \\ &\quad +\delta\mathbf{v}_{ij-1}\Delta t-\frac{1}{2}\Delta\tilde{\mathrm{R}}_{ij-1}\left(\tilde{\mathbf{a}}_{j-1}-\mathbf{b}_i^a\right)^\wedge\delta\boldsymbol{\phi}_{ij-1}\Delta t^2+\frac{1}{2}\Delta\tilde{\mathrm{R}}_{ij-1}\boldsymbol{\eta}_{j-1}^{ad}\Delta t^2 \\ &= \delta\mathbf{p}_{ij-1}+\delta\mathbf{v}_{ij-1}\Delta t-\frac{1}{2}\Delta\tilde{\mathrm{R}}_{ij-1}\left(\tilde{\mathbf{a}}_{j-1}-\mathbf{b}_i^a\right)^\wedge\delta\boldsymbol{\phi}_{ij-1}\Delta t^2+\frac{1}{2}\Delta\tilde{\mathrm{R}}_{ij-1}\boldsymbol{\eta}_{j-1}^{ad}\Delta t^2\end{aligned} \tag{C.61}$$

Recalling that $\boldsymbol{\eta}_{ik}^\Delta \doteq [\delta\boldsymbol{\phi}_{ik}, \delta\mathbf{v}_{ik}, \delta\mathbf{p}_{ik}]$, and defining the IMU measurement noise $\boldsymbol{\eta}_k^d \doteq [\boldsymbol{\eta}_k^{gd} \;\; \boldsymbol{\eta}_k^{ad}]$,[6] we can finally write Eqs. (C.59)-(C.61) in compact matrix form as:

$$\boldsymbol{\eta}_{ij}^\Delta = \mathbf{A}_{j-1}\boldsymbol{\eta}_{ij-1}^\Delta + \mathbf{B}_{j-1}\boldsymbol{\eta}_{j-1}^d, \tag{C.62}$$

From the linear model (C.62) and given the covariance $\boldsymbol{\Sigma}_\eta \in \mathbb{R}^{6\times6}$ of the raw IMU measurements noise $\boldsymbol{\eta}_k^d$, it is now possible to compute the preintegrated measurement covariance iteratively:

$$\boldsymbol{\Sigma}_{ij} = \mathbf{A}_{j-1}\boldsymbol{\Sigma}_{ij-1}\mathbf{A}_{j-1}^\mathsf{T} + \mathbf{B}_{j-1}\boldsymbol{\Sigma}_\eta\mathbf{B}_{j-1}^\mathsf{T} \tag{C.63}$$

starting from initial conditions $\boldsymbol{\Sigma}_{ii} = \mathbf{0}_{9\times9}$.

Bias Correction via First-Order Updates

In this section we provide a complete derivation of the first-order bias correction proposed in Section C.6.3.

[6] Both $\boldsymbol{\eta}_{ij}^\Delta$ and $\boldsymbol{\eta}_k^d$ are column vectors: we omit the transpose in the definition to keep notation simple.

Let us assume that we have computed the preintegrated variables at a given bias estimate $\bar{\mathbf{b}}_i \doteq [\bar{\mathbf{b}}_i^g \;\; \bar{\mathbf{b}}_i^a]$, and let us denote the corresponding preintegrated measurements as

$$\Delta\bar{\mathrm{R}}_{ij} \doteq \Delta\tilde{\mathrm{R}}_{ij}(\bar{\mathbf{b}}_i),\; \Delta\bar{\mathbf{v}}_{ij} \doteq \Delta\tilde{\mathbf{v}}_{ij}(\bar{\mathbf{b}}_i),\; \Delta\bar{\mathbf{p}}_{ij} \doteq \Delta\tilde{\mathbf{p}}_{ij}(\bar{\mathbf{b}}_i). \tag{C.64}$$

In this section we want to devise an expression to "update" $\Delta\bar{\mathrm{R}}_{ij}$, $\Delta\bar{\mathbf{v}}_{ij}$, $\Delta\bar{\mathbf{p}}_{ij}$ when our bias estimate changes.

Consider the case in which we get a new estimate $\hat{\mathbf{b}}_i \leftarrow \bar{\mathbf{b}}_i + \delta\mathbf{b}_i$, where $\delta\mathbf{b}_i$ is a *small* correction w.r.t. the previous estimate $\bar{\mathbf{b}}_i$.

We start with the bias correction for the preintegrated rotation measurement. The key idea here is to write $\Delta\tilde{\mathrm{R}}_{ij}(\hat{\mathbf{b}}_i)$ (the preintegrated measurement at the new bias estimate) as a function of $\Delta\bar{\mathrm{R}}_{ij}$ (the preintegrated measurement at the old bias estimate), "plus" a first-order correction. Recalling Eq. (C.35), we write $\Delta\tilde{\mathrm{R}}_{ij}(\hat{\mathbf{b}}_i)$ as:

$$\Delta\tilde{\mathrm{R}}_{ij}(\hat{\mathbf{b}}_i) = \prod_{k=i}^{j-1} \mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}_k - \hat{\mathbf{b}}_i^g\right)\Delta t\right) \tag{C.65}$$

Substituting $\hat{\mathbf{b}}_i = \bar{\mathbf{b}}_i + \delta\mathbf{b}_i$ in the previous expression and using the first-order approximation (C.4) in each factor (we assumed small $\delta\mathbf{b}_i$):

$$\begin{aligned}\Delta\tilde{\mathrm{R}}_{ij}(\hat{\mathbf{b}}_i) &= \prod_{k=i}^{j-1} \mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}_k - (\bar{\mathbf{b}}_i^g + \delta\mathbf{b}_i^g)\right)\Delta t\right) \\ &\simeq \prod_{k=i}^{j-1} \mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}_k - \bar{\mathbf{b}}_i^g\right)\Delta t\right) \mathrm{Exp}\left(-\mathrm{J}_r^k\, \delta\mathbf{b}_i^g\, \Delta t\right).\end{aligned} \tag{C.66}$$

Now, we rearrange the terms in the product, by "moving" the terms including $\delta\mathbf{b}_i^{gd}$ to the end, using the relation (C.11):

$$\Delta\tilde{\mathrm{R}}_{ij}(\hat{\mathbf{b}}_i) = \Delta\bar{\mathrm{R}}_{ij} \prod_{k=i}^{j-1} \mathrm{Exp}\left(-\Delta\tilde{\mathrm{R}}_{k+1j}(\bar{\mathbf{b}}_i)^{\mathsf{T}}\, \mathrm{J}_r^k\, \delta\mathbf{b}_i^g\, \Delta t\right), \tag{C.67}$$

where we used the fact that by definition it holds that $\Delta\bar{\mathrm{R}}_{ij} = \prod_{k=i}^{j-1} \mathrm{Exp}\left(\left(\tilde{\boldsymbol{\omega}}_k - \bar{\mathbf{b}}_i^g\right)\Delta t\right)$. Repeated application of the first-order approximation (C.7) (recall that $\delta\mathbf{b}_i^g$ is small, hence the right Jacobians are close to the identity) produces:

$$\begin{aligned}\Delta\tilde{\mathrm{R}}_{ij}(\hat{\mathbf{b}}_i) &\simeq \Delta\bar{\mathrm{R}}_{ij} \mathrm{Exp}\left(\sum_{k=i}^{j-1} -\Delta\tilde{\mathrm{R}}_{k+1j}(\bar{\mathbf{b}}_i)^{\mathsf{T}}\, \mathrm{J}_r^k \delta\mathbf{b}_i^g \Delta t\right) \\ &= \Delta\bar{\mathrm{R}}_{ij} \mathrm{Exp}\left(\frac{\partial\Delta\bar{\mathrm{R}}_{ij}}{\partial\mathbf{b}^g}\delta\mathbf{b}_i^g\right)\end{aligned} \tag{C.68}$$

Using (C.68) we can now update the preintegrated rotation measurement $\Delta\tilde{\mathbf{R}}_{ij}(\bar{\mathbf{b}}_i)$ to get $\Delta\tilde{\mathbf{R}}_{ij}(\hat{\mathbf{b}}_i)$ without repeating the integration.

Let us now focus on the bias correction of the preintegrated velocity $\Delta\tilde{\mathbf{v}}_{ij}(\hat{\mathbf{b}}_i)$:

$$
\begin{aligned}
\Delta\tilde{\mathbf{v}}_{ij}(\hat{\mathbf{b}}_i) &= \sum_{k=i}^{j-1} \Delta\tilde{\mathbf{R}}_{ik}(\hat{\mathbf{b}}_i)\left(\tilde{\mathbf{a}}_k - \bar{\mathbf{b}}_i^a - \delta\mathbf{b}_i^a\right)\Delta t \qquad \text{(C.69)}\\
&\overset{\text{(C.68)}}{\simeq} \sum_{k=i}^{j-1} \Delta\bar{\mathbf{R}}_{ik}\mathrm{Exp}\left(\frac{\partial\Delta\bar{\mathbf{R}}_{ik}}{\partial\mathbf{b}^g}\delta\mathbf{b}_i^g\right)\left(\tilde{\mathbf{a}}_k - \bar{\mathbf{b}}_i^a - \delta\mathbf{b}_i^a\right)\Delta t\\
&\overset{\text{(C.4)}}{\simeq} \sum_{k=i}^{j-1} \Delta\bar{\mathbf{R}}_{ik}\left(\mathbf{I} + \left(\frac{\partial\Delta\bar{\mathbf{R}}_{ik}}{\partial\mathbf{b}^g}\delta\mathbf{b}_i^g\right)^\wedge\right)\left(\tilde{\mathbf{a}}_k - \bar{\mathbf{b}}_i^a - \delta\mathbf{b}_i^a\right)\Delta t\\
&\overset{(a)}{\simeq} \Delta\bar{\mathbf{v}}_{ij} - \sum_{k=i}^{j-1}\Delta\bar{\mathbf{R}}_{ik}\Delta t\delta\mathbf{b}_i^a + \sum_{k=i}^{j-1}\Delta\bar{\mathbf{R}}_{ik}\left(\frac{\partial\Delta\bar{\mathbf{R}}_{ik}}{\partial\mathbf{b}^g}\delta\mathbf{b}_i^g\right)^\wedge\left(\tilde{\mathbf{a}}_k - \bar{\mathbf{b}}_i^a\right)\Delta t\\
&\overset{\text{(C.2)}}{=} \Delta\bar{\mathbf{v}}_{ij} - \sum_{k=i}^{j-1}\Delta\bar{\mathbf{R}}_{ik}\Delta t\delta\mathbf{b}_i^a - \sum_{k=i}^{j-1}\Delta\bar{\mathbf{R}}_{ik}\left(\tilde{\mathbf{a}}_k - \bar{\mathbf{b}}_i^a\right)^\wedge\frac{\partial\Delta\bar{\mathbf{R}}_{ik}}{\partial\mathbf{b}^g}\Delta t\,\delta\mathbf{b}_i^g\\
&= \Delta\bar{\mathbf{v}}_{ij} + \frac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^a}\delta\mathbf{b}_i^a + \frac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^g}\delta\mathbf{b}_i^g
\end{aligned}
$$

Where for (a), we used $\Delta\bar{\mathbf{v}}_{ij} = \sum_{k=i}^{j-1}\Delta\bar{\mathbf{R}}_{ik}\left(\tilde{\mathbf{a}}_k - \bar{\mathbf{b}}_i^a\right)\Delta t$. Exactly the same derivation can be repeated for $\Delta\tilde{\mathbf{p}}_{ij}(\hat{\mathbf{b}}_i)$. Summarizing, the Jacobians used for the a posteriori bias update in Eq. (C.44) are:

$$
\begin{aligned}
\frac{\partial\Delta\bar{\mathbf{R}}_{ij}}{\partial\mathbf{b}^g} &= -\textstyle\sum_{k=i}^{j-1}\left[\Delta\tilde{\mathbf{R}}_{k+1j}(\bar{\mathbf{b}}_i)^\mathsf{T}\ \mathrm{J}_r^k\ \Delta t\right]\\
\frac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^a} &= -\textstyle\sum_{k=i}^{j-1}\Delta\bar{\mathbf{R}}_{ik}\Delta t\\
\frac{\partial\Delta\bar{\mathbf{v}}_{ij}}{\partial\mathbf{b}^g} &= -\textstyle\sum_{k=i}^{j-1}\Delta\bar{\mathbf{R}}_{ik}\left(\tilde{\mathbf{a}}_k - \bar{\mathbf{b}}_i^a\right)^\wedge\frac{\partial\Delta\bar{\mathbf{R}}_{ik}}{\partial\mathbf{b}^g}\Delta t\\
\frac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}^a} &= \textstyle\sum_{k=i}^{j-1}\frac{\partial\Delta\bar{\mathbf{v}}_{ik}}{\partial\mathbf{b}^a}\Delta t - \frac{1}{2}\Delta\bar{\mathbf{R}}_{ik}\Delta t^2\\
\frac{\partial\Delta\bar{\mathbf{p}}_{ij}}{\partial\mathbf{b}^g} &= \textstyle\sum_{k=i}^{j-1}\frac{\partial\Delta\bar{\mathbf{v}}_{ik}}{\partial\mathbf{b}^g}\Delta t - \frac{1}{2}\Delta\bar{\mathbf{R}}_{ik}\left(\tilde{\mathbf{a}}_k - \bar{\mathbf{b}}_i^a\right)^\wedge\frac{\partial\Delta\bar{\mathbf{R}}_{ik}}{\partial\mathbf{b}^g}\Delta t^2
\end{aligned}
$$

Note that the Jacobians can be computed incrementally, as new measurements arrive.

Jacobians of Residual Errors

In this section we provide analytic expressions for the Jacobian matrices of the residual errors in Eq. (C.45). These Jacobians are crucial when using iterative optimization techniques (e.g., the Gauss-Newton method of Section C.3.3) to minimize the cost in Eq. (C.26).

"Lifting" the cost function (see Section C.3.3) consists in substituting the following

retractions:

$$\begin{array}{rclrcl} \mathtt{R}_i & \leftarrow & \mathtt{R}_i \operatorname{Exp}(\delta\boldsymbol{\phi}_i), & \mathtt{R}_j & \leftarrow & \mathtt{R}_j \operatorname{Exp}(\delta\boldsymbol{\phi}_j), \\ \mathbf{p}_i & \leftarrow & \mathbf{p}_i + \mathtt{R}_i\delta\mathbf{p}_i, & \mathbf{p}_j & \leftarrow & \mathbf{p}_j + \mathtt{R}_j\delta\mathbf{p}_j, \\ \mathbf{v}_i & \leftarrow & \mathbf{v}_i + \delta\mathbf{v}_i, & \mathbf{v}_j & \leftarrow & \mathbf{v}_j + \delta\mathbf{v}_i, \\ \delta\mathbf{b}_i^g & \leftarrow & \delta\mathbf{b}_i^g + \tilde{\delta}\mathbf{b}_i^g, & \delta\mathbf{b}_i^a & \leftarrow & \delta\mathbf{b}_i^a + \tilde{\delta}\mathbf{b}_i^a, \end{array} \tag{C.70}$$

The process of lifting makes the residual errors a function defined on a vector space, on which it is easy to compute Jacobians. Therefore, in the following sections we derive the Jacobians w.r.t. the vectors $\delta\boldsymbol{\phi}_i, \delta\mathbf{p}_i, \delta\mathbf{v}_i, \delta\boldsymbol{\phi}_j, \delta\mathbf{p}_j, \delta\mathbf{v}_j, \tilde{\delta}\mathbf{b}_i^g, \tilde{\delta}\mathbf{b}_i^a$.

Jacobians of $\mathbf{r}_{\Delta\mathbf{p}_{ij}}$

Since $\mathbf{r}_{\Delta\mathbf{p}_{ij}}$ is linear in $\delta\mathbf{b}_i^g$ and $\delta\mathbf{b}_i^a$, and the retraction is simply a vector sum, the Jacobians of $\mathbf{r}_{\Delta\mathbf{p}_{ij}}$ w.r.t. $\tilde{\delta}\mathbf{b}_i^g, \tilde{\delta}\mathbf{b}_i^a$ are simply the matrix coefficients of $\delta\mathbf{b}_i^g$ and $\delta\mathbf{b}_i^a$. Moreover, $\mathtt{R}_j$ and $\mathbf{v}_j$ do not appear in $\mathbf{r}_{\Delta\mathbf{p}_{ij}}$, hence the Jacobians w.r.t. $\delta\boldsymbol{\phi}_j, \delta\mathbf{v}_j$ are zero. Let us focus on the remaining Jacobians:

$$\begin{aligned} \mathbf{r}_{\Delta\mathbf{p}_{ij}}(\mathbf{p}_i + \mathtt{R}_i\delta\mathbf{p}_i) &= \mathtt{R}_i^\mathsf{T}\left(\mathbf{p}_j - \mathbf{p}_i - \mathtt{R}_i\delta\mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2\right) - C \\ &= \mathbf{r}_{\Delta\mathbf{p}_{ij}}(\mathbf{p}_i) + (-\mathbf{I}_{3\times 1})\delta\mathbf{p}_i \end{aligned} \tag{C.71}$$

$$\begin{aligned} \mathbf{r}_{\Delta\mathbf{p}_{ij}}(\mathbf{p}_j + \mathtt{R}_j\delta\mathbf{p}_j) &= \mathtt{R}_i^\mathsf{T}\left(\mathbf{p}_j + \mathtt{R}_j\delta\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2\right) - C \\ &= \mathbf{r}_{\Delta\mathbf{p}_{ij}}(\mathbf{p}_j) + (\mathtt{R}_i^\mathsf{T}\mathtt{R}_j)\delta\mathbf{p}_j \end{aligned} \tag{C.72}$$

$$\begin{aligned} \mathbf{r}_{\Delta\mathbf{p}_{ij}}(\mathbf{v}_i + \delta\mathbf{v}_i) &= \mathtt{R}_i^\mathsf{T}\left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \delta\mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2\right) - C \\ &= \mathbf{r}_{\Delta\mathbf{p}_{ij}}(\mathbf{v}_i) + (-\mathtt{R}_i^\mathsf{T}\Delta t_{ij})\delta\mathbf{v}_i \end{aligned} \tag{C.73}$$

$$\begin{aligned} \mathbf{r}_{\Delta\mathbf{p}_{ij}}(\mathtt{R}_i \operatorname{Exp}(\delta\boldsymbol{\phi}_i)) &= (\mathtt{R}_i\operatorname{Exp}(\delta\boldsymbol{\phi}_i))^\mathsf{T}\left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2\right) - C \\ &\overset{(C.4)}{\simeq} (\mathbf{I} - \delta\boldsymbol{\phi}_i^\wedge)\mathtt{R}_i^\mathsf{T}\left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2\right) - C \\ &\overset{(C.2)}{=} \mathbf{r}_{\Delta\mathbf{p}_{ij}}(\mathtt{R}_i) + \left(\mathtt{R}_i^\mathsf{T}\left(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2\right)\right)^\wedge \delta\boldsymbol{\phi}_i. \end{aligned} \tag{C.74}$$

Where we used the shorthand $C \doteq \Delta\tilde{\mathbf{p}}_{ij} + \frac{\partial \Delta\bar{\mathbf{p}}_{ij}}{\partial \mathbf{b}_i^g}\delta\mathbf{b}_i^g + \frac{\partial \Delta\bar{\mathbf{p}}_{ij}}{\partial \mathbf{b}_i^a}\delta\mathbf{b}_i^a$. Summarizing, the Jacobians of $\mathbf{r}_{\Delta\mathbf{p}_{ij}}$ are:

$$\begin{aligned}
\frac{\partial \mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial \delta\boldsymbol{\phi}_i} &= (\mathtt{R}_i^\mathsf{T}(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \tfrac{1}{2}\mathbf{g}\Delta t_{ij}^2))^\wedge & \frac{\partial \mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial \delta\boldsymbol{\phi}_j} &= \mathbf{0} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial \delta\mathbf{p}_i} &= -\mathbf{I}_{3\times 1} & \frac{\partial \mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial \delta\mathbf{p}_j} &= \mathtt{R}_i^\mathsf{T}\mathtt{R}_j \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial \delta\mathbf{v}_i} &= -\mathtt{R}_i^\mathsf{T}\Delta t_{ij} & \frac{\partial \mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial \delta\mathbf{v}_j} &= \mathbf{0} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial \delta\tilde{\mathbf{b}}_i^a} &= -\frac{\partial \Delta\bar{\mathbf{p}}_{ij}}{\partial \mathbf{b}_i^a} & \frac{\partial \mathbf{r}_{\Delta\mathbf{p}_{ij}}}{\partial \delta\tilde{\mathbf{b}}_i^g} &= -\frac{\partial \Delta\bar{\mathbf{p}}_{ij}}{\partial \mathbf{b}_i^g}
\end{aligned}$$

Jacobians of $\mathbf{r}_{\Delta\mathbf{v}_{ij}}$

As in the previous section, $\mathbf{r}_{\Delta\mathbf{v}_{ij}}$ is linear in $\delta\mathbf{b}_i^g$ and $\delta\mathbf{b}_i^a$, hence the Jacobians of $\mathbf{r}_{\Delta\mathbf{v}_{ij}}$ w.r.t. $\tilde{\delta}\mathbf{b}_i^g, \tilde{\delta}\mathbf{b}_i^a$ are simply the matrix coefficients of $\delta\mathbf{b}_i^g$ and $\delta\mathbf{b}_i^a$. Moreover, $\mathtt{R}_j$, $\mathbf{p}_i$, and $\mathbf{p}_j$ do not appear in $\mathbf{r}_{\Delta\mathbf{v}_{ij}}$, hence the Jacobians w.r.t. $\delta\boldsymbol{\phi}_j, \delta\mathbf{p}_i, \delta\mathbf{p}_j$ are zero. The remaining Jacobias are computed as:

$$\begin{aligned}
\mathbf{r}_{\Delta\mathbf{v}_{ij}}(\mathbf{v}_i + \delta\mathbf{v}_i) &= \mathtt{R}_i^\mathsf{T}\left(\mathbf{v}_j - \mathbf{v}_i - \delta\mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right) - D \\
&= \mathbf{r}_{\Delta\mathbf{v}}(\mathbf{v}_i) - \mathtt{R}_i^\mathsf{T}\delta\mathbf{v}_i
\end{aligned} \tag{C.75}$$

$$\begin{aligned}
\mathbf{r}_{\Delta\mathbf{v}_{ij}}(\mathbf{v}_j + \delta\mathbf{v}_j) &= \mathtt{R}_i^\mathsf{T}\left(\mathbf{v}_j + \delta\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right) - D \\
&= \mathbf{r}_{\Delta\mathbf{v}}(\mathbf{v}_j) + \mathtt{R}_i^\mathsf{T}\delta\mathbf{v}_j
\end{aligned} \tag{C.76}$$

$$\begin{aligned}
\mathbf{r}_{\Delta\mathbf{v}_{ij}}(\mathtt{R}_i\,\mathrm{Exp}(\delta\boldsymbol{\phi}_i)) &= (\mathtt{R}_i\,\mathrm{Exp}(\delta\boldsymbol{\phi}_i))^\mathsf{T}\left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right) - D \\
&\overset{(C.4)}{\simeq} (\mathbf{I} - \delta\boldsymbol{\phi}_i^\wedge)\mathtt{R}_i^\mathsf{T}\left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right) - D \\
&\overset{(C.2)}{=} \mathbf{r}_{\Delta\mathbf{v}}(\mathtt{R}_i) + \left(\mathtt{R}_i^\mathsf{T}\left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right)\right)^\wedge \delta\boldsymbol{\phi}_i,
\end{aligned} \tag{C.77}$$

with $D \doteq \left[\Delta\tilde{\mathbf{v}}_{ij} + \frac{\partial \Delta\bar{\mathbf{v}}_{ij}}{\partial \mathbf{b}_i^g}\delta\mathbf{b}_i^g + \frac{\partial \Delta\bar{\mathbf{v}}_{ij}}{\partial \mathbf{b}_i^a}\delta\mathbf{b}_i^a\right]$. Summarizing, the Jacobians of $\mathbf{r}_{\Delta\mathbf{v}_{ij}}$ are:

$$\begin{aligned}
\frac{\partial \mathbf{r}_{\Delta\mathbf{v}_{ij}}}{\partial \delta\boldsymbol{\phi}_i} &= \left(\mathtt{R}_i^\mathsf{T}\left(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}\right)\right)^\wedge & \frac{\partial \mathbf{r}_{\Delta\mathbf{v}_{ij}}}{\partial \delta\boldsymbol{\phi}_j} &= \mathbf{0} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{v}_{ij}}}{\partial \delta\mathbf{p}_i} &= \mathbf{0} & \frac{\partial \mathbf{r}_{\Delta\mathbf{v}_{ij}}}{\partial \delta\mathbf{p}_j} &= \mathbf{0} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{v}_{ij}}}{\partial \delta\mathbf{v}_i} &= -\mathtt{R}_i^\mathsf{T} & \frac{\partial \mathbf{r}_{\Delta\mathbf{v}_{ij}}}{\partial \delta\mathbf{v}_j} &= \mathtt{R}_i^\mathsf{T} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{v}_{ij}}}{\partial \delta\tilde{\mathbf{b}}_i^a} &= -\frac{\partial \Delta\bar{\mathbf{v}}_{ij}}{\partial \mathbf{b}_i^a} & \frac{\partial \mathbf{r}_{\Delta\mathbf{v}_{ij}}}{\partial \delta\tilde{\mathbf{b}}_i^g} &= -\frac{\partial \Delta\bar{\mathbf{v}}_{ij}}{\partial \mathbf{b}_i^g}
\end{aligned}$$

Jacobians of $\mathbf{r}_{\Delta \mathtt{R}_{ij}}$

The derivation of the Jacobians of $\mathbf{r}_{\Delta \mathtt{R}_{ij}}$ is slightly more involved. We first note that $\mathbf{p}_i, \mathbf{p}_j, \mathbf{v}_i, \mathbf{v}_j, \delta\mathbf{b}_i^a$ do not appear in the expression of $\mathbf{r}_{\Delta \mathtt{R}_{ij}}$, hence the corresponding Jacobians are zero. The remaining Jacobians can be computed as follows:

$$\begin{aligned}
\mathbf{r}_{\Delta \mathtt{R}_{ij}}(\mathtt{R}_i \operatorname{Exp}(\delta\boldsymbol{\phi}_i)) &= \operatorname{Log}\left(\left(\Delta\tilde{\mathtt{R}}_{ij}(\bar{\mathbf{b}}_i^g)E\right)^\mathsf{T} (\mathtt{R}_i \operatorname{Exp}(\delta\boldsymbol{\phi}_i))^\mathsf{T} \mathtt{R}_j \right) \\
&= \operatorname{Log}\left(\left(\Delta\tilde{\mathtt{R}}_{ij}(\bar{\mathbf{b}}_i^g)E\right)^\mathsf{T} \operatorname{Exp}(-\delta\boldsymbol{\phi}_i)\mathtt{R}_i^\mathsf{T}\mathtt{R}_j \right) \\
&\overset{(C.11)}{=} \operatorname{Log}\left(\left(\Delta\tilde{\mathtt{R}}_{ij}(\bar{\mathbf{b}}_i^g)E\right)^\mathsf{T} \mathtt{R}_i^\mathsf{T}\mathtt{R}_j \operatorname{Exp}(-\mathtt{R}_j^\mathsf{T}\mathtt{R}_i\delta\boldsymbol{\phi}_i) \right) \\
&\overset{(C.9)}{\simeq} \mathbf{r}_{\Delta\mathtt{R}}(\mathtt{R}_i) - \mathtt{J}_r^{-1}(\mathbf{r}_{\Delta\mathtt{R}}(\mathtt{R}_i))\mathtt{R}_j^\mathsf{T}\mathtt{R}_i\delta\boldsymbol{\phi}_i
\end{aligned} \tag{C.78}$$

$$\begin{aligned}
\mathbf{r}_{\Delta \mathtt{R}_{ij}}(\mathtt{R}_j \operatorname{Exp}(\delta\boldsymbol{\phi}_j)) &= \operatorname{Log}\left(\left(\Delta\tilde{\mathtt{R}}_{ij}(\bar{\mathbf{b}}_i^g)E\right)^\mathsf{T} \mathtt{R}_i^\mathsf{T}(\mathtt{R}_j \operatorname{Exp}(\delta\boldsymbol{\phi}_j)) \right) \\
&\overset{(C.9)}{\simeq} \mathbf{r}_{\Delta\mathtt{R}}(\mathtt{R}_j) + \mathtt{J}_r^{-1}(\mathbf{r}_{\Delta\mathtt{R}}(\mathtt{R}_j))\delta\boldsymbol{\phi}_j
\end{aligned} \tag{C.79}$$

$$\mathbf{r}_{\Delta \mathtt{R}_{ij}}(\delta\mathbf{b}_i^g + \tilde{\delta}\mathbf{b}_i^g) = \operatorname{Log}\left(\left(\Delta\tilde{\mathtt{R}}_{ij}(\bar{\mathbf{b}}_i^g) \operatorname{Exp}\left(\frac{\partial \Delta\bar{\mathtt{R}}_{ij}}{\partial \mathbf{b}^g}(\delta\mathbf{b}_i^g + \tilde{\delta}\mathbf{b}_i^g) \right) \right)^\mathsf{T} \mathtt{R}_i^\mathsf{T}\mathtt{R}_j \right) \tag{C.80}$$

$$\begin{aligned}
&\overset{(C.7)}{\simeq} \operatorname{Log}\left(\left(\Delta\tilde{\mathtt{R}}_{ij}(\bar{\mathbf{b}}_i^g)\ E \operatorname{Exp}\left(\mathtt{J}_r^b \frac{\partial \Delta\bar{\mathtt{R}}_{ij}}{\partial \mathbf{b}^g}\tilde{\delta}\mathbf{b}_i^g \right) \right)^\mathsf{T} \mathtt{R}_i^\mathsf{T}\mathtt{R}_j \right) \\
&= \operatorname{Log}\left(\operatorname{Exp}\left(-\mathtt{J}_r^b \frac{\partial \Delta\bar{\mathtt{R}}_{ij}}{\partial \mathbf{b}^g}\tilde{\delta}\mathbf{b}_i^g \right) \left(\Delta\tilde{\mathtt{R}}_{ij}(\bar{\mathbf{b}}_i^g)\ E\right)^\mathsf{T} \mathtt{R}_i^\mathsf{T}\mathtt{R}_j \right) \\
&= \operatorname{Log}\left(\operatorname{Exp}\left(-\mathtt{J}_r^b \frac{\partial \Delta\bar{\mathtt{R}}_{ij}}{\partial \mathbf{b}^g}\tilde{\delta}\mathbf{b}_i^g \right) \operatorname{Exp}\left(\mathbf{r}_{\Delta\mathtt{R}_{ij}}(\delta\mathbf{b}_i^g) \right) \right) \\
&\overset{(C.11)}{=} \operatorname{Log}\left(\operatorname{Exp}\left(\mathbf{r}_{\Delta\mathtt{R}_{ij}}(\delta\mathbf{b}_i^g) \right) \cdot \operatorname{Exp}\left(-\operatorname{Exp}\left(\mathbf{r}_{\Delta\mathtt{R}_{ij}}(\delta\mathbf{b}_i^g) \right)^\mathsf{T} \mathtt{J}_r^b \frac{\partial \Delta\bar{\mathtt{R}}_{ij}}{\partial \mathbf{b}^g}\tilde{\delta}\mathbf{b}_i^g \right) \right) \\
&\overset{(C.9)}{\simeq} \mathbf{r}_{\Delta\mathtt{R}_{ij}}(\delta\mathbf{b}_i^g) - \mathtt{J}_r^{-1}\left(\mathbf{r}_{\Delta\mathtt{R}_{ij}}(\delta\mathbf{b}_i^g) \right) \operatorname{Exp}\left(\mathbf{r}_{\Delta\mathtt{R}_{ij}}(\delta\mathbf{b}_i^g) \right)^\mathsf{T} \mathtt{J}_r^b \frac{\partial \Delta\bar{\mathtt{R}}_{ij}}{\partial \mathbf{b}^g}\ \tilde{\delta}\mathbf{b}_i^g,
\end{aligned}$$

where we used the shorthands $E \doteq \text{Exp}\left(\frac{\partial \Delta\bar{\mathbf{R}}_{ij}}{\partial \mathbf{b}^g}\delta\mathbf{b}^g\right)$ and $\mathtt{J}_r^b \doteq \mathtt{J}_r\left(\frac{\partial \Delta\bar{\mathbf{R}}_{ij}}{\partial \mathbf{b}^g}\delta\mathbf{b}_i^g\right)$. In summary, the Jacobians of $\mathbf{r}_{\Delta\mathbf{R}_{ij}}$ are:

$$\begin{aligned}
\frac{\partial \mathbf{r}_{\Delta\mathbf{R}_{ij}}}{\partial \delta\boldsymbol{\phi}_i} &= -\mathtt{J}_r^{-1}(\mathbf{r}_{\Delta\mathbf{R}}(\mathbf{R}_i))\mathbf{R}_j^\mathsf{T}\mathbf{R}_i & \frac{\partial \mathbf{r}_{\Delta\mathbf{R}_{ij}}}{\partial \delta\mathbf{p}_i} &= \mathbf{0} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{R}_{ij}}}{\partial \delta\mathbf{v}_i} &= \mathbf{0} & \frac{\partial \mathbf{r}_{\Delta\mathbf{R}_{ij}}}{\partial \delta\boldsymbol{\phi}_j} &= \mathtt{J}_r^{-1}(\mathbf{r}_{\Delta\mathbf{R}}(\mathbf{R}_j)) \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{R}_{ij}}}{\partial \delta\mathbf{p}_j} &= \mathbf{0} & \frac{\partial \mathbf{r}_{\Delta\mathbf{R}_{ij}}}{\partial \delta\mathbf{v}_j} &= \mathbf{0} \\
\frac{\partial \mathbf{r}_{\Delta\mathbf{R}_{ij}}}{\partial \delta\tilde{\mathbf{b}}_i^a} &= \mathbf{0} & \frac{\partial \mathbf{r}_{\Delta\mathbf{R}_{ij}}}{\partial \delta\tilde{\mathbf{b}}_i^g} &= \alpha
\end{aligned} \tag{C.81}$$

with $\alpha = -\mathtt{J}_r^{-1}\left(\mathbf{r}_{\Delta\mathbf{R}_{ij}}(\delta\mathbf{b}_i^g)\right)\text{Exp}\left(\mathbf{r}_{\Delta\mathbf{R}_{ij}}(\delta\mathbf{b}_i^g)\right)^\mathsf{T}\mathtt{J}_r^b\frac{\partial \Delta\bar{\mathbf{R}}_{ij}}{\partial \mathbf{b}^g}$.

Structureless Vision Factors: Null Space Projection

In this section we provide a more efficient implementation of the structureless vision factors, described in Section C.7.

Let us consider Eq. (C.55). Recall that $\mathbf{Q} \doteq (\mathbf{I} - \mathbf{E}_l(\mathbf{E}_l^\mathsf{T}\mathbf{E}_l)^{-1}\mathbf{E}_l^\mathsf{T}) \in \mathbb{R}^{2n_l \times 2n_l}$ is an *orthogonal projector* of $\mathbf{E}_l$, where n_l is the number of cameras observing landmark l. Roughly speaking, $\mathbf{Q}$ projects any vector in $\mathbb{R}^{2n_l}$ to the null space of the matrix $\mathbf{E}_l$. Since $\mathbf{E}_l \in \mathbb{R}^{2n_l \times 3}$ has rank 3, the dimension of its null space is $2n_l - 3$. Any basis $\mathbf{E}_l^\perp \in \mathbb{R}^{2n_l \times 2n_l - 3}$ of the null space of $\mathbf{E}_l$ satisfies the following relation Meyer [2000]:

$$\mathbf{E}_l^\perp\left((\mathbf{E}_l^\perp)^\mathsf{T}\mathbf{E}_l^\perp\right)^{-1}(\mathbf{E}_l^\perp)^\mathsf{T} = \mathbf{I} - \mathbf{E}_l(\mathbf{E}_l^\mathsf{T}\mathbf{E}_l)^{-1}\mathbf{E}_l^\mathsf{T}. \tag{C.82}$$

A basis for the null space can be easily computed from $\mathbf{E}_l$ using SVD. Such basis is *unitary*, i.e., satisfies $(\mathbf{E}_l^\perp)^\mathsf{T}\mathbf{E}_l^\perp = \mathbf{I}$. Substituting (C.82) into (C.55), and recalling that $\mathbf{E}_l^\perp$ is a unitary matrix, we obtain:

$$\begin{aligned}
&\sum_{l=1}^{L} \|\mathbf{E}_l^\perp(\mathbf{E}_l^\perp)^\mathsf{T}\left(\mathbf{F}_l\,\delta\mathbf{T}_{\mathcal{X}(l)} - \mathbf{b}_l\right)\|^2 \\
&= \sum_{l=1}^{L}\left(\mathbf{E}_l^\perp(\mathbf{E}_l^\perp)^\mathsf{T}(\mathbf{F}_l\,\delta\mathbf{T}_{\mathcal{X}(l)} - \mathbf{b}_l)\right)^\mathsf{T}\left(\mathbf{E}_l^\perp(\mathbf{E}_l^\perp)^\mathsf{T}(\mathbf{F}_l\,\delta\mathbf{T}_{\mathcal{X}(l)} - \mathbf{b}_l)\right) \\
&= \sum_{l=1}^{L}\left(\mathbf{F}_l\,\delta\mathbf{T}_{\mathcal{X}(l)} - \mathbf{b}_l\right)^\mathsf{T}\mathbf{E}_l^\perp\overbrace{(\mathbf{E}_l^\perp)^\mathsf{T}\mathbf{E}_l^\perp}^{=\mathbf{I}_{3\times 3}}(\mathbf{E}_l^\perp)^\mathsf{T}\left(\mathbf{F}_l\,\delta\mathbf{T}_{\mathcal{X}(l)} - \mathbf{b}_l\right) \\
&= \sum_{l=1}^{L}\|(\mathbf{E}_l^\perp)^\mathsf{T}\left(\mathbf{F}_l\,\delta\mathbf{T}_{\mathcal{X}(l)} - \mathbf{b}_l\right)\|^2
\end{aligned} \tag{C.83}$$

which is an alternative representation of the cost function (C.55). This representation

is usually preferable from a computational standpoint, as it does not include matrix inversion and can be computed using a smaller number of matrix multiplications.

Rotation Rate Integration Using Euler Angles

In this section, we recall how to integrate rotation rate measurements using the Euler angle parametrization. Let $\tilde{\boldsymbol{\omega}}_k$ be the rotation rate measurement at time k and $\boldsymbol{\eta}_k^g$ be the corresponding noise. Then, given the vector of Euler angles at time k, namely $\boldsymbol{\theta}_k \in \mathbb{R}^3$, we can integrate the rotation rate measurement $\tilde{\boldsymbol{\omega}}_k$ and get $\boldsymbol{\theta}_{k+1}$ as follows:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + [E'(\boldsymbol{\theta}_k)]^{-1}(\tilde{\boldsymbol{\omega}}_k - \boldsymbol{\eta}_k^g)\Delta t, \tag{C.84}$$

where the matrix $E'(\boldsymbol{\theta}_k)$ is the *conjugate Euler angle rate matrix* Diebel [2006]. The covariance of $\boldsymbol{\theta}_{k+1}$ can be approximated by a first-order propagation as:

$$\boldsymbol{\Sigma}_{k+1}^{\text{Euler}} = \mathbf{A}_k \boldsymbol{\Sigma}_k^{\text{Euler}} \mathbf{A}_k^\mathsf{T} + \mathbf{B}_k \boldsymbol{\Sigma}_{\boldsymbol{\eta}} \mathbf{B}_k^\mathsf{T} \tag{C.85}$$

where $\mathbf{A}_k \doteq \mathbf{I}_{3\times 3} + \frac{\partial [E'(\boldsymbol{\theta}_k)]^{-1}}{\partial \boldsymbol{\theta}_k}\Delta t$, $\mathbf{B}_k = -[E'(\boldsymbol{\theta}_k)]^{-1}\Delta t$, and $\boldsymbol{\Sigma}_{\boldsymbol{\eta}}$ is the covariance of the measurement noise $\boldsymbol{\eta}_k^{gd}$.

D Probabilistic, Monocular Dense Reconstruction

M. Pizzoli, C. Forster, and D. Scaramuzza. REMODE: Probabilistic, monocular dense reconstruction in real time. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2609–2616, 2014. URL http://dx.doi.org/10.1109/ICRA.2014.6907233.

REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time

Matia Pizzoli, Christian Forster, Davide Scaramuzza

Abstract — In this paper, we solve the problem of estimating dense and accurate depth maps from a single moving camera. A probabilistic depth measurement is carried out in real time on a per-pixel basis and the computed uncertainty is used to reject erroneous estimations and provide live feedback on the reconstruction progress. Our contribution is a novel approach to depth map computation that combines Bayesian estimation and recent development on convex optimization for image processing. We demonstrate that our method outperforms state-of-the-art techniques in terms of accuracy, while exhibiting high efficiency in memory usage and computing power. We call our approach REMODE (REgularized MOnocular Depth Estimation) and the CUDA-based implementation runs at 30Hz on a laptop computer.

Introduction

We present a method to compute an accurate, three-dimensional reconstruction of the scene observed by a moving camera and provide, in real time, information about the progress and the reliability of the ongoing estimation process. This problem is highly relevant in robot perception, where cameras are valuable and widespread sensors. From a single moving camera, it is possible to collect appearance and range information about the observed three-dimensional scene. In a multi-view stereo setting, the uncertainty on the depth measurement depends on the noise affecting image formation, on the camera poses, and the scene structure. Knowing how these factors affect the measurement uncertainty, it is possible to achieve arbitrarily high levels of confidence by collecting measurements from different vantage points. Such a capability is particularly valuable in robotics. For instance, if the camera is mounted on a robotic arm, the available high level of mobility can be exploited to disambiguate scene details and occlusions at a wide range of distances. The monocular setting is also an appealing sensing modality

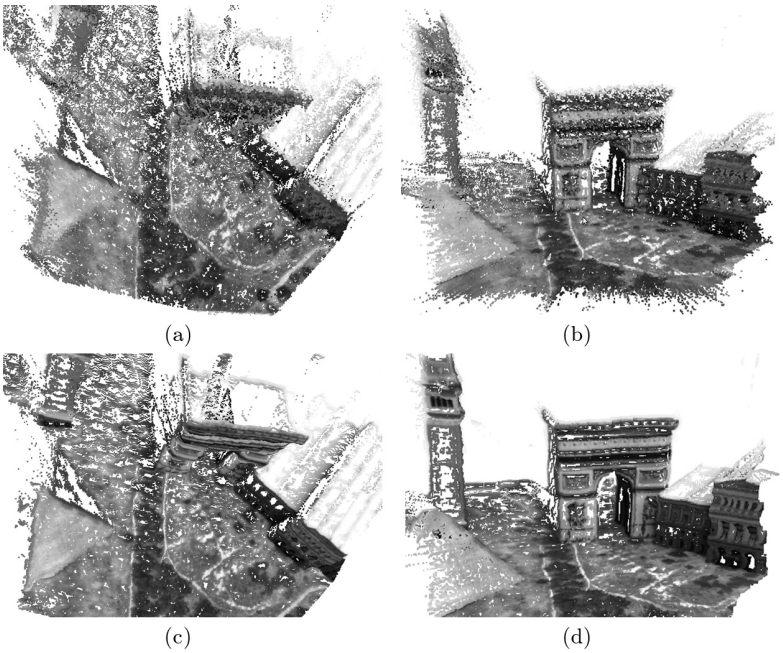


Figure D.1 – In monocular dense reconstructions, the probabilistic approach to depth estimation produces compact and efficient representations. Highly parallelizable implementations are achieved by estimating the depth for every pixel independently. A smoothing step is nonetheless required to achieve robustness against noise and mitigate the effect of erroneous measurements. Figures (a) and (b) show the result of Bayesian depth estimation from multiple views; (c) and (d) show the same result after the de-noising step that we propose in this paper.

for Micro Aerial Vehicles (MAVs), where strict limitations apply on payload and power consumption. In this case, the high agility turns the platform into a formidable depth sensor, able to deal with a wide depth range and capable of achieving arbitrarily high confidence in the measurement. Inevitably, this high flexibility comes at a cost. The pose of the camera must be known and its accuracy influences the reconstruction quality. For a camera, information resides in the changing of the intensity gradient and this modality naturally fails in presence of low informative scenes that produce untextured images. It is therefore crucial to know how reliable each measurement is.

Related Work

The problem of reconstructing the scene from images collected by a moving camera has been studied for more than two decades and is known as Structure from Motion in computer vision [Hartley and Zisserman, 2003] and Monocular SLAM in robotics [Davison, 2003]. The growing interest for dense reconstructions has renewed the attention in multi-view stereo techniques [Matthies et al., 1988, Kang et al., 2001, Seitz et al., 2006, Furukawa and Ponce, 2010], where the involved computational complexity used to prevent applications in robot perception. In robotics, the use of

RGBD cameras is favouring the development of techniques for highly-detailed [Meilland and Comport, 2013] and spatially-extended reconstructions [Whelan et al., 2013], their applicability being limited to short range measurements and indoor environments. The literature in dense stereo is vast and we refer to [Hirschmuller and Scharstein, 2009] for a comparison. However, few relevant works have addressed real-time, dense reconstruction from a single moving camera and they shed light on some important aspects. Figure D.1 illustrates the problem we address in this paper. If, on one hand, estimating the depth independently for every pixel leads to efficient, parallel implementations, on the other hand the authors of [Gallup et al., 2007, Stühmer et al., 2010, Newcombe et al., 2011b] argued that, similar to other computer vision problems, such as image de-noising [Rudin et al., 1992] and optical flow estimation [Werlberger et al., 2010], a smoothing step is required in order to deal with noise and spurious measurements. In [Stühmer et al., 2010], smoothness priors were enforced over the reconstructed scene by minimizing a regularized energy functional based on aggregating a photometric cost over different depth hypothesis and penalizing non-smooth surfaces. The authors showed that the integration of multiple images leads to significantly higher robustness to noise. A similar argument is put forth in [Newcombe et al., 2011b], where the advantage of photometric cost aggregation [Szeliski and Scharstein, 2004] over a large number of images taken from nearby viewpoints is demonstrated. Regularized energy functionals also play an important role in recent methods for volumetric reconstruction [Zach, 2008, Graber et al., 2011, Forster et al., 2013], where the three-dimensional surface of a scene is generated by fusing several depth maps obtained from multi-view stereo. Depending on the scene appearance and the used stereo baselines, the computed depth maps are potentially noisy and a robust fusion method helps mitigate the effect of wrong depth estimations.

However, despite the ground-breaking results, these approaches present some limitations when addressing tasks in robot perception. Equally weighting measurements from small and large baselines, in close and far scenes, causes the aggregated cost to frequently present multiple or no minima. Depending on the depth range and sampling, these failures are not always recoverable by the subsequent optimization step. Furthermore, an inadequate number of images can lead to a poorly constrained initialization for the optimization and erroneous measurements that are hard to detect. It is not clear how many images should be collected, depending on the motion of the camera and the scene structure. Finally, the number of depth hypotheses controls the computational complexity, and the applicability is, thus, limited to scenes bounded in depth.

Contributions and Outline

The discussed limitations are overcome by probabilistic approaches handling measurement uncertainty. A compact representation and a Bayesian depth estimation from

multi-view stereo were proposed in [Vogiatzis and Hernández, 2011]. We build on their results for per-pixel depth estimation and introduce an optimization step to enforce spatial regularity over the recovered depth map. We propose a regularization term based on the weighted Huber norm but, differently from [Newcombe et al., 2011b], we use the depth uncertainty to drive the smoothing and exploit a convex formulation for which a highly parallelizable solution scheme has been recently introduced [Chambolle and Pock, 2011]. The contributions of this paper are the following:

- a probabilistic depth map, in which the Bayesian scheme in [Vogiatzis and Hernández, 2011] is integrated in a monocular SLAM algorithm to estimate per-pixel depths based on the live camera stream;
- a fast smoothing method that takes into account the measurement uncertainty to provide spatial regularity and mitigates the effect of noisy camera localization.

The outline of the paper follows. In Section D.2 we detail our method for depth estimation from monocular views and in Section D.3 we provide the implementation details. Section D.4 is dedicated to the discussion on the experimental evaluation. Finally, in Section D.5, we summarize our contribution and draw the conclusion.

Monocular Dense Reconstruction

Considerations

The solution we propose to compute a dense reconstruction from a single moving camera is motivated by the following considerations.

A measure of uncertainty is needed in robotic perception many reconstruction pipelines previously proposed in computer vision and graphics literature aim at providing visually appealing maps. In contrast, we are interested in accurately mapping the environment in order to allow robotic tasks, such as autonomous navigation and exploration, active perception or situation awareness in the case of human-operated systems. As a passive sensing modality, measurement uncertainty in monocular multi-view stereo is related to the camera motion and the amount of visual information present in the scene (e.g. texture). A probabilistic depth map handles measure uncertainty, thus, allowing efficient updating, optimal sensor placement, and fusion with different sensors.

A dense reconstruction is needed to interact sparse visual maps based on image features have been successfully used in robotics, e.g. to solve the SLAM problem.

However, feature definitions change between sensing modalities and tasks; dense representations are, thus, required to actually solve the problem of registering data among largely different vantage points based on the three-dimensional structure [Forster et al., 2013]. When the task involves physical interaction with the environment—as in obstacle avoidance, path planning and manipulation—the highest achievable level of detail is desirable in order to estimate the surfaces involved in the interaction.

Perception must be fast differently from many state-of-the-art systems, in order to be useful in robot perception our pipeline must run in real-time using the robot's on-board computing power. Depth estimation must be updated efficiently and the uncertainty in the estimation must improve according to the information conveyed by the image and the current camera pose.

In the designing of the monocular multi view stereo algorithm, these considerations naturally bring to the formulation of the following requirements: depth estimation must take into account the uncertainty arising from the scene and the camera pose and the estimation must be carried out on-line and updated sequentially. Bayesian estimation offers a natural way to deal with measure uncertainty, to handle sequential measurement updates and to reject unreliable estimations in an on-line fashion.

Depthmap from Multi View Stereo

We formulate the depth computation as a Bayesian estimation problem. Each observation provides a depth measurement by triangulating from the reference view and the last acquired view. The depth of a pixel is described by a parametric model that is updated on the basis of the current observation. Finally, smoothness on the resulting depth map is enforced by minimizing a regularized energy functional.

Bayesian Estimation

Let the rigid body transformation $\mathbf{T}_{k,w} \in SE(3)$ describe the pose of the camera acquiring the k-th view, i.e., $\mathbf{T}_{k,w}$ transforms scene points ${}_w p \in \mathbb{R}^3$ from the world frame to the frame of the k-th camera pose: ${}_k\mathbf{p} = \mathtt{T}_{k,w}\ {}_w\mathbf{p}$.

We denote the intensity image collected from the k-th camera pose as $I_k : \Omega \subset \mathbb{R}^2 \mapsto \mathbb{R}$, where Ω is the image domain. We denote by $\mathbf{u} \in \Omega$ a point in image coordinates.

An observation is a pair $\{I_k, \mathtt{T}_{k,w}\}$. A sequence of n observations is identified by the sequence of time steps $k = r, \ldots, r+n$, in which the r-th observation is taken as reference. A depth hypothesis d_k is generated from the observation $\{I_k, \mathtt{T}_{k,w}\}$ by triangulating $\mathbf{u}$ from the views r and k.

The sequence of d_k for $k = r, \ldots, r+n$ denotes a set of noisy depth measurements. We model the depth sensor as a distribution that mixes a good measurement (normally distributed around the true depth $\hat{d}$) and an outlier measurement (uniformly distributed in an interval $[d_{min}, d_{max}]$ which is known to contain the depth for the structure of interest):

$$p(d_k|\hat{d}, \rho) = \rho\mathcal{N}(d_k|\hat{d}, \tau_k^2) + (1-\rho)\mathcal{U}(d_k|d_{min}, d_{max}), \tag{D.1}$$

where ρ and τ_k^2 are the probability and the variance of a good measurement, respectively. Assuming independent observations, the Bayesian estimation for $\hat{d}$ on the basis of the measurements $d_{r+1}, \ldots, d_k$ is given by the posterior

$$p(\hat{d}, \rho|d_{r+1}, \ldots, d_k) \propto p(\hat{d}, \rho) \prod_k p(d_k|\hat{d}, \rho), \tag{D.2}$$

with $p(\hat{d}, \rho)$ being a prior on the true depth and the ratio of good measurements supporting it. A sequential update is implemented by using the estimation at time step $k-1$ as a prior to combine with the observation at time step k. To this purpose, the authors of [Vogiatzis and Hernández, 2011] show that the posterior in (D.2) can be approximated by the product of a Gaussian distribution for the depth and a Beta distribution for the inlier ratio:

$$q(\hat{d}, \rho|a_k, b_k, \mu_k, \sigma_k^2) = Beta(\rho|a_k, b_k)\mathcal{N}(\hat{d}|\mu_k, \sigma_k^2), \tag{D.3}$$

where a_k and b_k are the parameters controlling the Beta distribution. The choice is motivated by the fact that the $Beta \times Gaussian$ is the approximating distribution minimizing the Kullback-Leibler divergence from the true posterior (D.2). Upon the k-th observation, the update takes the form

$$p(\hat{d}, \rho|d_{r+1}, \ldots, d_k) \approx q(\hat{d}, \rho|a_{k-1}, b_{k-1}, \mu_{k-1}, \sigma_{k-1}^2)\ p(d_k|\hat{d}, \rho)\ const \tag{D.4}$$

and the authors of [Vogiatzis and Hernández, 2011] approximated the true posterior (D.4) with a $Beta \times Gaussian$ distribution by matching the first and second order moments for $\hat{d}$ and ρ. The updates formulas for a_k, b_k, μ_k and σ_k^2 are thus derived and we refer to the original work in [Vogiatzis and Hernández, 2011] for the details on the derivation.

Regularized Posterior

We now detail our solution to the problem of smoothing the depth map $D(\mathbf{u})$. For every pixel $\mathbf{u} \in \Omega$, the depth estimation and its confidence upon the k-th observation are given, respectively, by μ_k and σ_k^2 in (D.3). We formulate the problem of computing

a de-noised depth map $F(\mathbf{u})$ as the following minimization:

$$\min_{F} \int_{\Omega} \{G(\mathbf{u}) \left\|\nabla F(\mathbf{u})\right\|_{\epsilon} + \lambda \left\|F(\mathbf{u}) - D(\mathbf{u})\right\|_{1}\} d\mathbf{u}, \tag{D.5}$$

where λ is a free parameter controlling the trade-off between the data term and the regularizer, and $G(\mathbf{u})$ is a weighting function related to the "G-Weighted Total Variation", introduced in [Bresson et al., 2007] in the context of image segmentation. We penalize non-smooth surfaces by making use of a regularization term based on the Huber norm of the gradient, defined as:

$$\left\|\nabla F(\mathbf{u})\right\|_{\epsilon} = \begin{cases} \frac{||\nabla F(\mathbf{u})||_2^2}{2\epsilon} & \text{if } ||\nabla F(\mathbf{u})||_2 \leq \epsilon, \\ ||\nabla F(\mathbf{u})||_1 - \frac{\epsilon}{2} & \text{otherwise .} \end{cases} \tag{D.6}$$

We chose the Huber norm because it allows smooth reconstruction while preserving discontinuities at strong depth gradient locations ([Newcombe et al., 2011b]). The weighting function $G(\mathbf{u})$ influences the strength of the regularization and we propose to compute it on the basis of the measure confidence for $\mathbf{u}$:

$$G(\mathbf{u}) = \mathbb{E}_{\rho}[q](\mathbf{u}) \frac{\sigma^2(\mathbf{u})}{\sigma^2_{max}} + \left\{1 - \mathbb{E}_{\rho}[q](\mathbf{u})\right\}, \tag{D.7}$$

where we have extended the notation for the expected value of the inlier ratio $\mathbb{E}_{\rho}[q]$ and the variance σ^2 in (D.3) to account for the specific pixel $\mathbf{u}$. The weighting function (D.7) affects the strength of the regularization term: for measurements with a high expected value for the inlier ratio ρ the weight is controlled by the measurement variance σ^2; measurements characterized by a small variance (i.e. reliable measurements) will be less affected by the regularization; differently, the contribution of the regularization term will be heavier for measurements characterized by a small expected value for the inlier ratio or higher measurement variance.

The solution to the minimization problem (D.5) is computed iteratively based on the work in [Chambolle and Pock, 2011]. The algorithm exploits the primal dual formulation of (D.5),

$$\min_{F} \max_{F^*} \langle \text{diag}(G)\nabla F, F^* \rangle + \lambda ||C - D||_1 - \delta_{F^*}(F^*) - \frac{\epsilon}{2}||F^*||_2^2,$$

and proceeds by alternating gradient descent and ascent steps in the primal and dual variables, namely F and F^*. The indicator function $\delta_{F^*}(F^*)$ is such that, for each F^*, $\delta_{F^*}(F^*) = 0$ if $||F^*||_1 \leq 1$, and otherwise ∞. Let t and t^* be the time steps for the gradient descent-ascent with respect to the primal and dual variable. The update steps in the case of the Weighted-Huber de-noising model (D.5) take the form

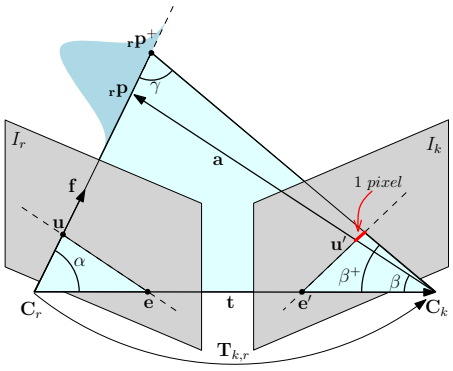


Figure D.2 – Computation of the measurement uncertainty. The camera poses acquiring the views I_r and I_k are related by the transformation $\mathrm{T}_{k,r}$. The camera centres C_r, C_k and the current estimation of the scene point ${}_r\mathbf{p}$ lie on the epipolar plane.The variance corresponding to one pixel along the epipolar line passing through $\mathbf{e}'$ and $\mathbf{u}'$ is computed as $\tau_k^2 = (||{}_r\mathbf{p}^+|| - ||{}_r\mathbf{p}||)^2$.

$$
\begin{aligned}
F_{n+1}^* &= \text{prox}\left(\frac{F_n^* + t^*(\text{diag}(G)\nabla)\bar{F}}{1 + t^*\epsilon}\right), \\
F_{n+1} &= \text{shrink}\left(F_n - t\left(\nabla^T \text{diag}(G)\right)F_{n+1}^*\right), \\
\bar{F}_{n+1} &= 2F_{n+1} - F_n,
\end{aligned}
\tag{D.8}
$$

where the resolvent operators are

$$
\text{prox}(\tilde{f}^*) = \frac{\tilde{f}^*}{\max(1, |\tilde{f}^*|)}, \qquad \text{shrink}(\tilde{f}) = \begin{cases} \tilde{f} - t\lambda & \text{if } \tilde{f} - d > t\lambda \\ \tilde{f} + t\lambda & \text{if } \tilde{f} - d < -t\lambda \\ d & \text{if } |\tilde{f} - d| \leq t\lambda \end{cases} \tag{D.9}
$$

and d is the noisy depth value corresponding to a specific pixel.

Implementation Details

The monocular reconstruction pipeline is designed to run in real time on a commodity laptop, using a CPU and a GPU. The proposed probabilistic depth map and convex optimization lead to highly parallel algorithms and we based our implementation on CUDA[1].

[1] `http://www.nvidia.com`

Camera pose estimation

At every time step k, the pose of the camera $\mathbf{T}_{k,r}$ in the depth map reference frame r is computed by a visual odometry routine that is based on recent advancement on semi-direct methods for camera localization [Forster et al., 2014b]. The algorithm operates directly on the image intensity, eliminating the need for costly feature extraction and resulting in sub-pixel accuracy at high frame-rates. Three-dimensional map points are estimated making use of the probabilistic method described in Section D.2.2, which proved at the same time highly robust, accurate and computationally efficient. Our implementation is characterized by an average drift in pose of 0.0038 metres per second for an average depth of 1 metre and a computing time of 3.3 milliseconds per acquired image on the experimental platform detailed in Section D.4. The visual odometry algorithm is run by the CPU, and its accuracy and efficiency support the simultaneous execution of the monocular reconstruction pipeline.

Measurement update

The parametric model in (D.3) is a compact representation, as it stores our confidence in the depth measurement corresponding to a pixel in only four parameters: a, b, μ and σ. When a reference frame is taken, the estimation for every pixel is initialized and updated with every subsequent view. We set the initial parameters $a_0 = 10$, $b_0 = 10$, $\mu_0 = 0.5(d_{min} + d_{max})$ and $\sigma_0 = \sigma_{max}$, where σ_{max} is such that 99% of the probability mass lies in the interval $[d_{min}, d_{max}]$. Upon the acquisition of the k-th view, the update introduced in [Vogiatzis and Hernández, 2011] is performed for every pixel of the reference view. We perform the update until the depth estimation converges or diverges. At this point, we can either consider the measurement reliable or discard it. We check the convergence and divergence conditions by looking at the variance of the depth posterior σ_k^2 and the estimated inlier ratio ρ_k. Let η_{inlier} and $\eta_{outlier}$ be thresholds on the estimated inlier ratio and σ_{thr} be a threshold on the variance of the depth posterior. We have three cases:

- if $\mathbb{E}_\rho[q] > \eta_{inlier}$ and $\sigma_k^2 < \sigma_{thr}^2$, then the estimation has converged;
- else if $\mathbb{E}_\rho[q] < \eta_{outlier}$, then the estimation has diverged;
- otherwise, the estimation continues.

The parameters η_{inlier}, $\eta_{outlier}$ and σ_{thr} control the estimation convergence and can be set according to the accuracy and robustness requirements for the application at hand.

In order to deal with higher depth ranges, we base our implementation on the inverse depth [Civera et al., 2008] and use the currently estimated variance to limit the search for correspondence on the epipolar line.

Measurement uncertainty

When triangulating matched points to estimate the depth from multiple views, frames taken from nearby vantage points are less affected by occlusions and allow high quality matches. On the other hand, a large baseline enables a more reliable depth estimation but with a higher chance to incur in occluded regions.

Referring to Figure D.2, let ${}_r\mathbf{p}$ be the current estimation of the scene point corresponding to the pixel $\mathbf{u}$ in the image I_r. The variance on the position of ${}_r\mathbf{p}$ is obtained by back-projecting a constant variance of one pixel in the image I_k. Let $\mathbf{t}$ be the translation component of $\mathtt{T}_{k,r}$ and $\mathbf{f} = \frac{{}_r\mathbf{p}}{||{}_r\mathbf{p}||}$, then

$$\mathbf{a} =_r \mathbf{p} - \mathbf{t}, \qquad \alpha = \arccos\left(\frac{\mathbf{f}\cdot\mathbf{t}}{||\mathbf{t}||}\right), \qquad \beta = \arccos\left(-\frac{\mathbf{a}\cdot\mathbf{t}}{||\mathbf{a}||\cdot||\mathbf{t}||}\right). \tag{D.10}$$

Let f be the camera focal length. The angle spanning one pixel can be added to β in order to compute γ and, thus, by applying the law of sines, recover the norm of ${}_r\mathbf{p}^+$:

$$\beta^+ = \beta + 2\tan^{-1}\left(\frac{1}{2f}\right) \qquad \gamma = \pi - \alpha - \beta^+ \qquad ||{}_r\mathbf{p}^+|| = ||\mathbf{t}||\frac{\sin\beta^+}{\sin\gamma}. \tag{D.11}$$

Therefore, the measurement uncertainty is computed as

$$\tau_k^2 = \left(||{}_r\mathbf{p}^+|| - ||{}_r\mathbf{p}||\right)^2. \tag{D.12}$$

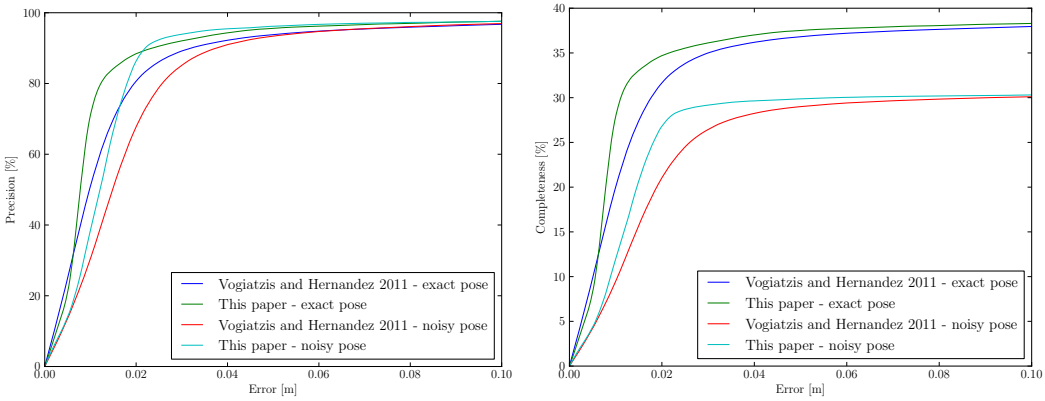
Experimental Evaluation

The platform we used for the experimental evaluation of the proposed monocular reconstruction method is an Intel i7-3720QM based laptop, equipped with 15 GB of RAM, and an NVIDIA Quadro K2000M GPU with 384 CUDA cores.

We chose the dataset presented in [Handa et al., 2012] in order to quantitatively evaluate our approach. The dataset consists of views generated through ray-tracing from a three-dimensional synthetic model. Along with each view, the related exact camera pose and depth maps are made available. Table D.1 summarizes the details for the sequences used in the evaluation.

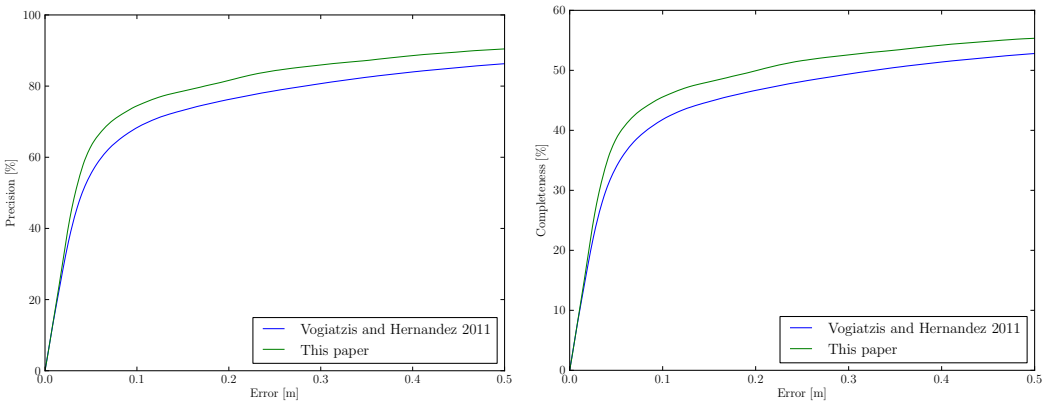
Table D.1 – Datasets for comparison against ground truth.

	Frames	Range [m]	Mean [m]	Motion [m]	Speed [m/s]
Over table	200	0.827-2.84	1.531	4.576	0.686
Fast motion	900	0.971-6.802	2.015	21.6	1.61



(a) (b)

Figure D.3 – Quantitative evaluation on the *over table* sequence. In (a) the precision is plotted, namely the percentage of converged estimations that are within a certain error from the ground truth. In (b) the completeness is plotted, namely the percentage of ground truth measurements that are within a certain error from the converged estimations.



(a) (b)

Figure D.4 – Quantitative evaluation on the *fast motion* sequence. In (a) the precision is plotted, namely the percentage of converged estimations that are within a certain error from the ground truth. In (b) the completeness is plotted, namely the percentage of ground truth measurements that are within a certain error from the converged estimations.

Over table identifies a sequence of views collected down-looking on a desktop scenario. The sequence is characterized by a frame rate of 30 frames per second and smooth camera motion. The sequence identified as *fast motion* is a collection of views generated at 60 frames per second with large and sudden changes of vantage point. The evaluation is based on comparison with the ground truth depth map corresponding to the view taken as reference in the reconstruction process. Two depth maps are compared by computing the sum of the per-pixel absolute difference. Since we are interested in evaluating the depth measurements that have been identified as reliable by our algorithm, we only take into account those measurements that have converged according to Section D.3.2. We therefore use the converged measurements to create the masks (e) in

Figure D.8 and Figure D.9, which are used in the comparison. We define two evaluation metrics: *precision*, namely the percentage of converged measurements that fall below a certain error when compared to the relative ground truth, and *completeness*, namely the percentage of ground truth depths that have been estimated by the proposed method within a certain error. In order to show the effectiveness of our approach, we compare our results with depth maps computed according to the state-of-the-art method introduced in [Vogiatzis and Hernández, 2011]. This work is at the basis of our probabilistic treatment and, so far, its applicability has been demonstrated only for reconstruction of small objects. For our comparison using the ground truth sequences, the parameters defining reliable measures have been set at $\eta_{inlier} = 0.6$, $\eta_{outlier} = 0.05$ and $\sigma_{thr} = \sigma_{max}/10^3$. The parameters governing the optimization were set at $\epsilon = 10^{-4}$ and $\lambda = 0.3$, and 200 iterations of the primal-dual update in (D.8) were run.

Figure D.3 reports the result of the evaluation on the *over table* sequence. Our approach is capable to recover a number of erroneous depth estimations, thus yielding a sensible improvement in terms of accuracy and completeness. To verify the robustness against noisy camera pose estimation, we corrupted the camera position with Gaussian noise, with zero mean and one centimetre standard deviation on each coordinate. The results show that the completeness drops. This is inevitable due to the smaller number of converged estimations. However, the computation of the depth map takes advantage of the de-noising step. This trend is even more evident in the *fast motion* sequence, depicted in Figure D.9. Here, according to the results in Figure D.4, the advantage of our approach is clearly demonstrated in terms of both precision and completeness. Handling measurement uncertainty, the probabilistic treatment of depth allows us to select the optimal trade-off between precision and accuracy by varying the σ_{thr} parameter. Figure D.5 shows how, for a given error tolerance, the completeness varies as a function of the variance σ^2 that characterizes a reliable measurement. We can see, for instance, that using a threshold $\sigma_{thr} = 6 \times 10^{-4}$, which is approximately 2×10^3 times the initialization value σ_{max}, more than 60% of the depth measurements computed by our method are affected by an error up to 15 centimetres, that is approximately 2.6% of the full depth range.

In order to demonstrate the effectiveness of the proposed approach on real time reconstructions, we present our results on the *City of Sights* stage set [Gruber et al.,

Table D.2 – Computing time for the evaluation data

	Update time [s]		Optimization time [s]	
	Mean	Variance	Mean	Variance
Over table	0.0382	0.0025	0.1107	0
Fast motion	0.0499	0.0035	0.1149	0
Live acquisition	0.0301	0.0011	0.1122	0.0044

2010]. We computed a point cloud from different depth maps acquired by a single hand-held camera. Our reconstruction pipeline was fed with images and camera poses computed by the underlying visual odometry (cfr. Section D.3.1) at 30 frames per second.

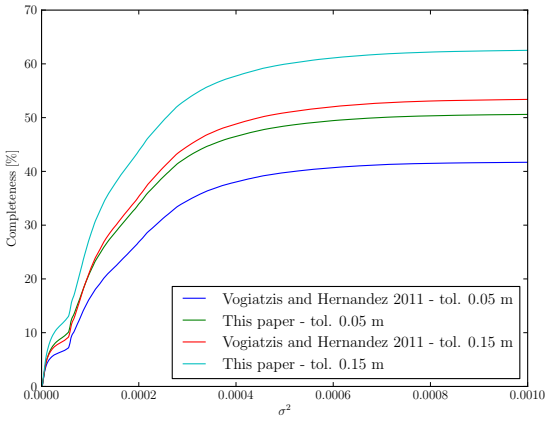
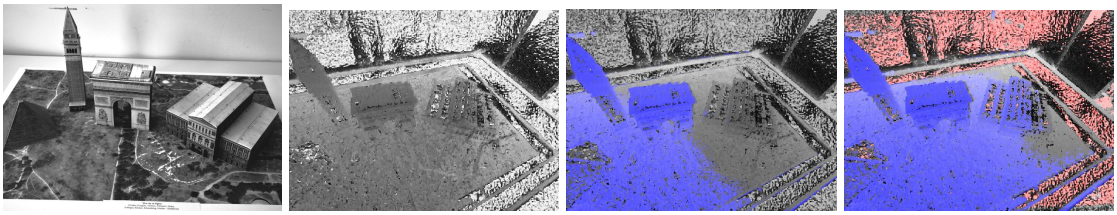


Figure D.5 – The percentage of ground truth measurements that are within an error of 5 and 15 centimetres is plotted as a function of the measurement variance σ^2.

Figure D.6 depicts the process of a live depth map acquisition. During the reconstruction, the convergence and divergence of estimations are displayed as a live feedback for the user (blue and red respectively in the figure), guiding the motion of the camera to acquire portions of the scene for which the estimation has not yet converged or diverged. A qualitative evaluation of the results can be drawn from Figures D.6 and D.7. The minimization in (E.3) imposes a smoothness constraint on the resulting surface and acts as a prior when the estimation is uncertain. Wrong depth computations, caused by shadows or matching errors (see Figure D.6b), cause the respective estimations to diverge (red points in Figure D.6d). The de-noising step propagates the depth value produced by converged measurements to those neighbours yielding low confidence, which are characterized by diverged measurements. The final result, in the form of a coloured point cloud rendered from two different viewpoints, is depicted in Figure D.7.

Finally, the proposed method is suitable for real time execution, as can be seen in Table D.2, where we have reported the computing time for the evaluation sequences. The computational cost of the proposed method is dominated by the search for correspondences on the epipolar line. When the motion of the camera is smooth, like in the cases of the *over table* dataset and live acquisition, the region selected for the search is small; when the camera motion forms large baselines, then the candidate search area is wider, affecting the computing time as in the case of the *fast motion* dataset. The depth range characterizing the volume of interest for the reconstruction also plays an important role, as the measurement uncertainty is higher for distant points (cfr. Section D.3.3). This causes the depth estimation to require a larger number of views to converge. Nonetheless, the estimation update runs in real time on the live 30 fps camera stream, for a camera resolution of 752×480 pixels. The computational cost of the optimization step depends only on the image size and number of iterations, and is thus constant among an evaluation sequence. Optimization was run several times during the *live acquisition*, triggered by the instantiation of new reference frames, while for the ground truth sequences the single optimization step that is performed motivates



(a) (b) (c) (d)

Figure D.6 – Depth map computation for the *City of Sights* stage set [Gruber et al., 2010]. Dark points are close, bright points are far. Blue and red identify converged and diverged estimations, respectively.

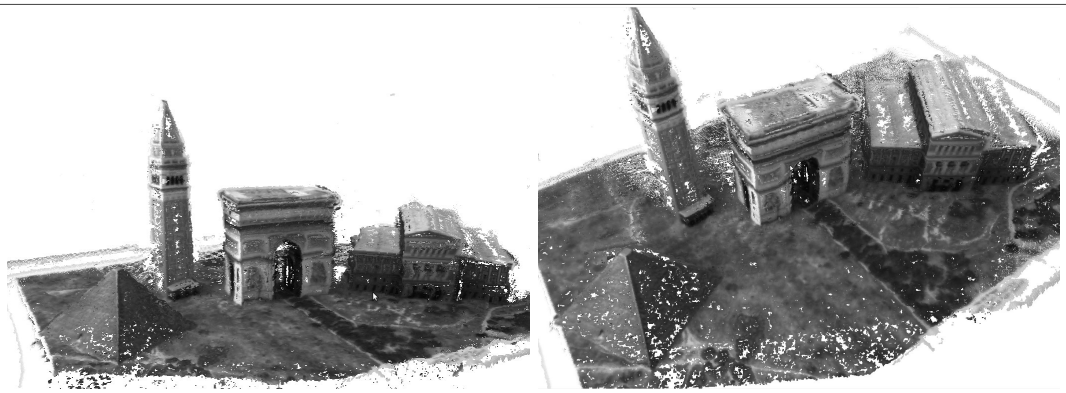


Figure D.7 – Reconstructed point clouds for the *City of Sights* stage set [Gruber et al., 2010].

the 0 variance entries in Table D.2.

A video demonstrating the reconstruction of scenes acquired by a hand-held camera and a flying robot, is available at the website http://rpg.ifi.uzh.ch/research_dense.html.

Conclusion

In this paper we presented REMODE, a probabilistic approach to monocular dense reconstruction for robot perception. Our method computes depth maps by combining Bayesian estimation and recent developments in convex optimization for image processing. We showed how a probabilistic update scheme can produce a compact and efficient representation of a depth map and its related uncertainty. In order to achieve real time execution on a live camera stream, we parallelized the computation of a depth map by considering each pixel independently. Afterwards, we introduced a fast smoothing step that takes into account the measurement uncertainty to enforce spatial regularity and mitigates the effect of noisy camera localization. We evaluated our method in terms of accuracy and completeness, showing a sensible improvement with respect to the current state-of-the-art. By handling measurement uncertainty,

our method provides real time information about the progress and the reliability of the ongoing reconstruction process. This information is highly valuable to drive the reconstruction, that is, to determine what views are most informative for the task at hand.

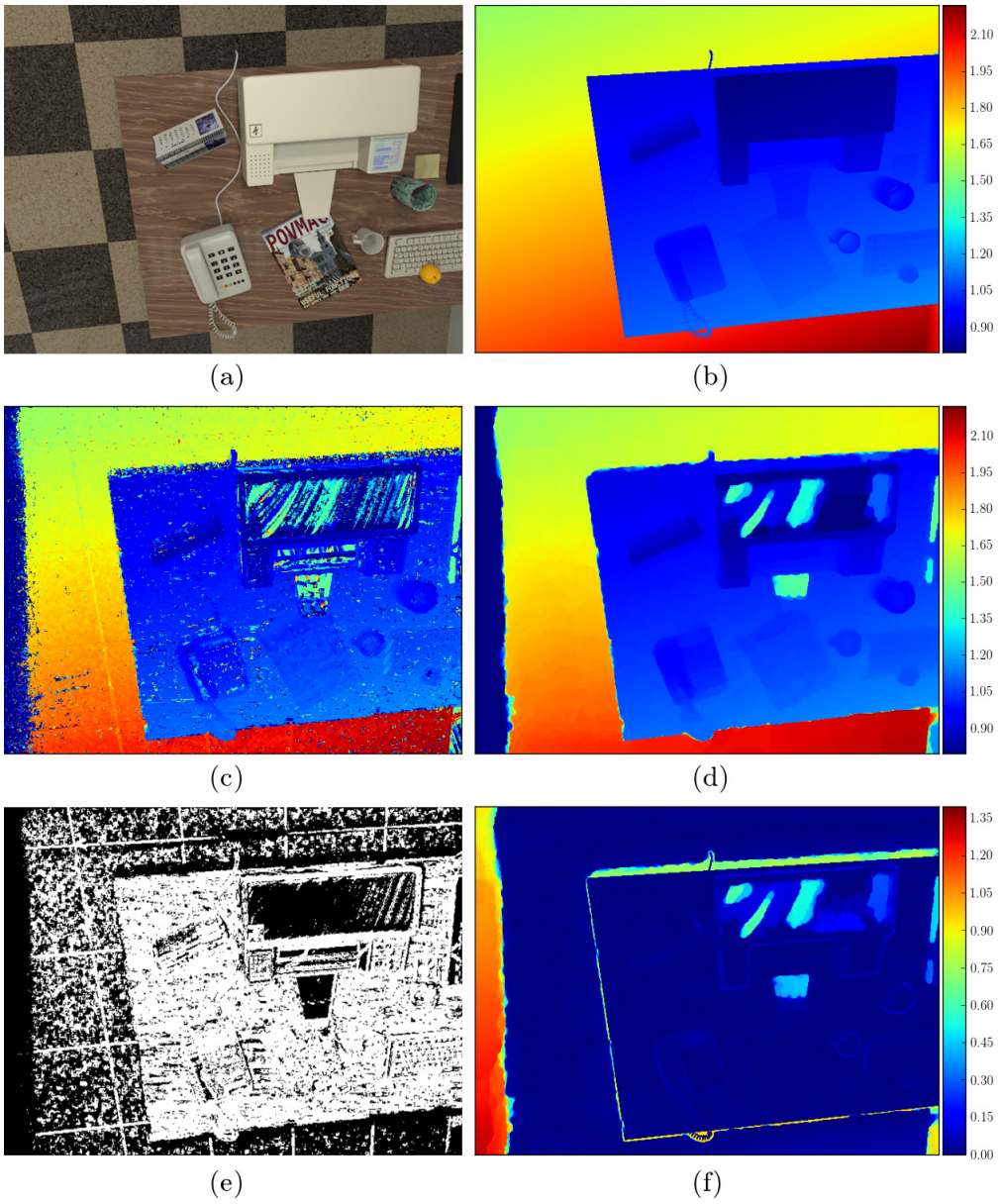


Figure D.8 – The *over table* evaluation sequence. (a): the reference view. (b): ground truth depth map. (c): depth map based on [Vogiatzis and Hernández, 2011]. (d): depth map computed by the proposed method. (e): map of reliable measurement according to Section D.3.2. (f): error for the proposed method.

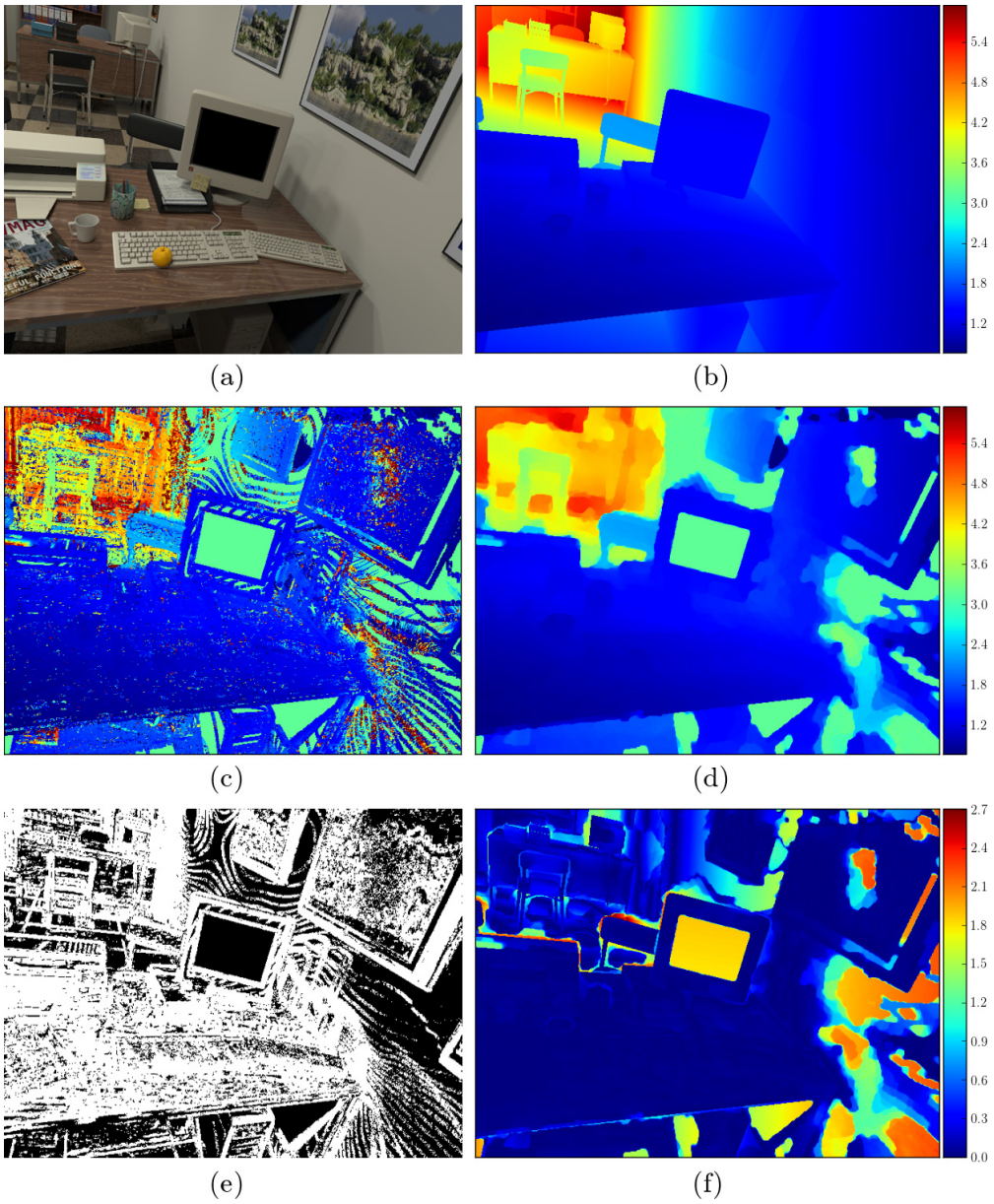


Figure D.9 – The *fast motion* evaluation sequence. (a): the reference view. (b): ground truth depth map. (c): depth map based on [Vogiatzis and Hernández, 2011]. (d): depth map computed by the proposed method. (e): map of reliable measurement according to Section D.3.2. (f): error for the proposed method.

E Dense Elevation Mapping

C. Forster, M. Faessler, F. Fontana, M. Werlberger, and D. Scaramuzza. Continuous on-board monocular-vision–based aerial elevation mapping for quadrotor landing. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 111–118, 2015. URL http://dx.doi.org/10.1109/ICRA.2015.7138988.

Continuous On-Board Monocular-Vision–based Elevation Mapping Applied to Autonomous Landing of Micro Aerial Vehicles

Christian Forster, Matthias Faessler, Flavio Fontana,
Manuel Werlberger, Davide Scaramuzza

Abstract — In this paper, we propose a resource-efficient system for real-time 3D terrain reconstruction and landing-spot detection for micro aerial vehicles. The system runs on an on-board smartphone processor and requires only the input of a single downlooking camera and an inertial measurement unit. We generate a two-dimensional elevation map that is probabilistic, of fixed size, and robot-centric, thus, always covering the area immediately underneath the robot. The elevation map is continuously updated at a rate of 1 Hz with depth maps that are triangulated from multiple views using recursive Bayesian estimation. To highlight the usefulness of the proposed mapping framework for autonomous navigation of micro aerial vehicles, we successfully demonstrate fully autonomous landing including landing-spot detection in real-world experiments.

Introduction

Autonomous Micro Aerial Vehicles (MAVs) will soon play a major role in industrial inspection, agriculture, search and rescue and consumer goods delivery. For autonomous operations in these fields, it is crucial that the vehicle is at all times fully aware of the surface immediately underneath: First, during normal operation, the vehicle should maintain a minimum distance to the ground surface to avoid crashing. Second, for autonomous landing, the MAV needs to identify, approach and land at a safe site without human intervention. Knowing the ground surface in previously-unknown

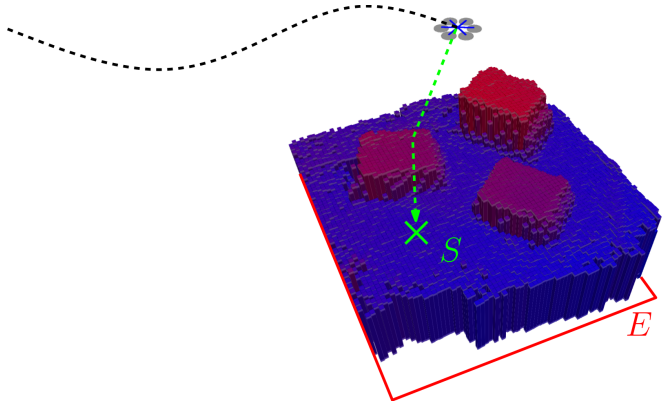


Figure E.1 – Illustration of the local elevation map E. The two-dimensional probabilistic grid map is of fixed size and centered below the MAV's position. The MAV updates the map continuously at a rate of 1 Hz using only the on-board smartphone processor and data from a single down-looking camera. The map enables the robot to autonomously detect and approach a landing spot S at any time (green trajectory).

environments is invaluable in case of forced landings due to emergencies like communication loss as well as for planned landings for, e.g., saving energy during monitoring operations or for the delivery of goods.

Large-scale Unmanned Aerial Vehicles (UAVs) often use range sensors to detect hazards, avoid obstacles or to land autonomously [Johnson et al., 2002, Scherer et al., 2012]. However, these active sensors are expensive, heavy and quickly drain the battery when used on small MAVs. Instead, given efficient and robust computer vision algorithms, active range sensors can be replaced by a single downward-looking camera. This setup is lightweight, cost effective, and, as shown in previous works [Scaramuzza et al., 2014], allows accurate localization and stabilization of the MAV in GPS denied environments, such as indoors, close to buildings, or below bridges.

In contrast to stereo and RGBD-cameras with fixed baselines, a single moving camera may be seen as a stereo setting that can dynamically adjust its baseline according to the required measurement accuracy as well as the structure and texture of the scene. Indeed, a single moving camera represents the most general setting for stereo vision. This property was previously exploited for real-time 3D terrain reconstruction from aerial images in order to detect landing spots [Johnson et al., 2005, Bosch et al., 2006, Desaraju et al., 2014] or for visualization purposes [Weiss et al., 2011a, Wendel et al., 2012, Pizzoli et al., 2014, Faessler et al., 2015].

In this paper, we propose a system for mapping the local ground environment underneath an MAV equipped with a single camera. Detailed dense and textured reconstructions are valuable for human operators and can be computed in real-time but off-board by streaming images to a ground station as shown in [Wendel et al.,

2012, Pizzoli et al., 2014, Faessler et al., 2015]. In order to work also for emergency maneuvers or autonomous flying in remote areas without the availability of a ground station, we restrict the system to solely use the computing capability on-board the MAV. To achieve this objective we utilize a coarse two-dimensional elevation map [Fankhauser et al., 2014] as on-board map representation, which is sufficient for many autonomous maneuvers in outdoor environments. A further advantage compared to other map representations, such as surface meshes [Weiss et al., 2011a], is the regular sampling of the surface and the possibility to fuse multiple elevation measurements via a probabilistic representation. The proposed system runs continuously on an on-board smartphone processor and updates the robot-centric elevation map of fixed dimension at a rate of 1 Hz. The system does not require any prior information of the scene or external navigation aids such as GPS.

Related Work

Real-time dense reconstruction with a single camera has been previously demonstrated in [Gallup et al., 2007, Stühmer et al., 2010, Newcombe et al., 2011b, Wendel et al., 2012, Pizzoli et al., 2014]. However, all previous approaches rely on heavy GPU parallelization and therefore can currently not be computed with the on-board computing power of an MAV. In [Pizzoli et al., 2014] we presented the REMODE (regularized monocular depth estimation) algorithm and demonstrated live but off-board dense mapping from an MAV. Therefore, we streamed on-board pose estimates provided by an accurate visual odometry algorithm [Forster et al., 2014b] together with images at a rate of 10 Hz to a ground station that was equipped with a powerful laptop computer and was capable to compute dense depth maps in real-time. In the current paper, we utilize REMODE to build a 2D elevation map and present modifications to the algorithm to run it on a smartphone CPU on-board the MAV.

Early works on vision-based autonomous landing for Unmanned Aerial Vehicles (UAV) were based on detecting known planar shapes (e.g., helipads with "H" markings) in images [Saripalli et al., 2002] or on the analysis of textures in single images [Garcia-Pardo et al., 2002]. Later works (e.g., [Johnson et al., 2005, Bosch et al., 2006, Desaraju et al., 2014]) assessed the risk of a landing spot by evaluating the roughness and inclination of the surface using 3D terrain reconstruction from images.

The first demonstration of vision based autonomous landing in unknown and hazardous terrain is described in [Johnson et al., 2005]. Similar to our work, structure-from-motion was used to estimate the relative pose of two monocular images and subsequently, a dense elevation map with a resolution of 19×27 cells was computed by matching and triangulating 600 regularly sampled features. The evaluation of the roughness and slope of the computed terrain map resulted in a binary classification of safe and hazardous landing areas. While this work detects the landing spot entirely

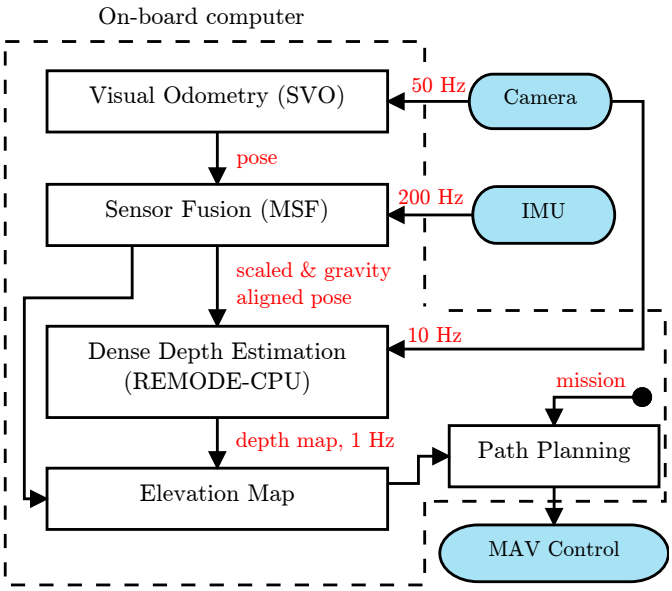


Figure E.2 – Overview of the main components and connections in the proposed system. All modules are running on-board the MAV.

based on two selected images, we continuously make depth measurements and fuse them in a local elevation map.

In [Bosch et al., 2006], homography estimation was used to compute the motion of the camera as well as to recover planar surfaces in the scene. Similar to our work, a probabilistic two-dimensional grid was used as map representation. However, the grid stored the probability of the cells being flat and not the actual elevation value as in our approach, therefore, barring the possibility to use the map for obstacle avoidance.

While previously mentioned works were passive in the sense that the exploration flight was pre-programmed, recent work [Desaraju et al., 2014] was *actively* choosing the best trajectory to explore and verify a landing spot. Due to computational complexity, the full system could not run entirely on-board in real-time. Thus, outdoor experiments were processed on datasets. In contrast to our work, only two frames were used to compute dense motion stereo, hence a criterion, based on the visibility of features and the inter-frame baseline, was needed to select two images. The probabilistic depth estimation in our work not only allows using every image for robust incremental estimation but also provides a measure of uncertainty that can be used for planning trajectories minimizing the uncertainty as fast as possible [Forster et al., 2014a].

Contributions

The contribution of this work is a monocular-vision–based 3D terrain scanning system that runs in real-time and continuously on a smartphone processor on-board an MAV. Therefore, we introduce a novel robot-centric elevation map representation to the MAV research community. To highlight the usefulness of the proposed elevation map, we demonstrate both indoor and outdoor experiments of a fully integrated landing spot detection and autonomous landing system for a lightweight quadrotor.

System Overview

Figure E.2 illustrates the proposed systems' main components and their linkage:

We use our *Semi-direct Visual Odometry (SVO)* [Forster et al., 2014b] to estimate the current MAV's pose given the image stream from the single downward-looking camera.[1] However, with a single camera we can obtain the relative camera motion only up to an unknown scale factor.

Therefore, in order to align the pose correctly with the gravity vector, and to estimate the scale of the trajectory, we fuse the output of SVO with the data coming from the on-board inertial measurement unit (IMU). For integrating the IMU's data, we use the *MSF (multi-sensor fusion)* software package [Lynen et al., 2013], which utilizes an extended Kalman filter.[2] Next, we compute depth estimates with a modified version of the *REMODE (REgularized MOnocular Depth Estimation)* [Pizzoli et al., 2014] algorithm. Details on the modifications of the REMODE algorithm for computing probabilistic depth maps purely relying on the on-board computing capability are given in Section E.3.

The generated depth maps are then used to incrementally update a *2D robot-centric elevation map* [Fankhauser et al., 2014]. Since the elevation map is probabilistic, we perform a Bayesian update step for the elevation values of the affected cells, whenever a new depth map is available. In addition, the elevation map moves together with the robot's pose as it is local and robot-centric. More details on the update steps and how to incorporate the depth measurements are given in Section E.4.

The flight trajectory of the MAV is provided by the *path planning* module which can be implemented in different ways: For instance, it has a pre-programmed flight path, or it obtains way-points from a remote operator, or it uses active vision in order to select the next-best views to make the current depth map converge as fast as possible. For further details on the active vision approach, we refer to [Forster et al., 2014a].

[1] Available at http://github.com/uzh-rpg/rpg_svo
[2] Available at https://github.com/ethz-asl/ethzasl_msf

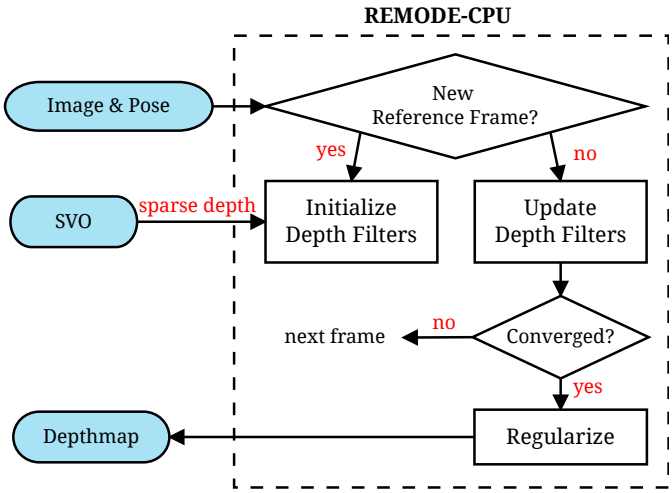


Figure E.3 – Overview of the monocular dense reconstruction system.

As an exemplar application of the given system, we show an autonomous landing of the MAV. Therefore, an additional module for *landing-spot detection* based on the elevation map is presented in Section E.5.

Monocular Dense Reconstruction

In the following, we summarize the REMODE (REgularized MOnocular Depth Estimation) algorithm, which we introduced in [Pizzoli et al., 2014], and describe the necessary modifications to run the algorithm in real-time on a smartphone processor on-board the MAV.

An overview of the algorithm is given in Figure E.3. The algorithm computes a dense depth map for selected reference views. The depth computation for a single pixel is formalized as a Bayesian estimation problem. Therefore, a so called *depth filter* is initialized for all pixels in every newly selected reference image $\mathtt{I}_r$ (see Figure E.4). Every subsequent image $\mathtt{I}_k$ is used to perform a recursive Bayesian update step of the depth estimates. The selection of reference frames — hence the amount of generated depth maps given a sequence of images — is based on two criterions: a new reference view is selected whenever (1) the uncertainties of the given depth estimates are below a certain threshold (thus the depth map has converged), or (2) when the spatial distance between the current camera pose and the reference view is larger than a certain threshold. After the depth map converged, we enforce its smoothness by applying a Total Variation (TV) based image filter.

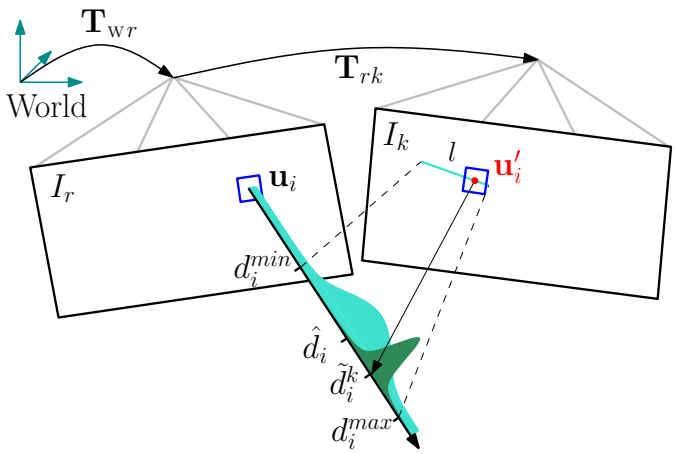


Figure E.4 – Probabilistic depth estimate $\hat{\rho}_i$ for pixel i in the reference frame r. The point at the true depth projects to similar image regions in both images (blue squares). Thus, the depth estimate is updated with the triangulated depth $\tilde{\rho}_i^k$ computed from the point $\mathbf{u}'_i$ of highest correlation with the reference patch. The point of highest correlation lies always on the epipolar line in the new image.

Depth Filter

Given a new reference frame $\mathtt{I}_r$, a depth filter is initialized for every pixel with a high uncertainty in depth and a mean that is derived from the sparse 3D reconstruction in the visual odometry (see Section E.3.3). The depth filter is described by a parametric model that is updated on the basis of every subsequent frame k.

Let the rigid body transformation $\mathbf{T}_{wr} \in SE(3)$ describe the pose of a reference frame r relative to the world frame W. Given a new observation $\{\mathtt{I}_k, \mathtt{T}_{wk}\}$, we project the 95% depth-confidence interval $[\rho_i^{\min}, \rho_i^{\max}]$ of the depth filter corresponding to pixel i into the image $\mathtt{I}_k$ and find a segment of the epipolar line l (see Figure E.4). Using the zero-mean sum of squared differences (ZMSSD) score on a 8×8 patch, we search the pixel $\mathbf{u}'_i$ on the epipolar line segment l that has highest correlation with the reference pixel $\mathbf{u}_i$. A depth measurement $\tilde{\rho}_i^k$ is generated from the observation by triangulating $\mathbf{u}_i$ and $\mathbf{u}'_i$ from the views r and k respectively. As proposed in [Vogiatzis and Hernández, 2011], we can model the measurements with a model that mixes "good" measurements (i.e., inliers) with "bad" ones (i.e., outliers). With probability ρ_i, the measurement is a good one and is normally distributed around the correct depth ρ_i with a measurement variance ${\tau_i^k}^2$. With probability $1 - \rho_i$, the measurement is an outlier and is uniformly distributed in an interval $[\rho_i^{\min}, \rho_i^{\max}]$):

$$p(\tilde{\rho}_i^k | \rho_i, \rho_i) = \rho_i \mathcal{N}(\tilde{\rho}_i^k | \rho_i, {\tau_i^k}^2) + (1 - \rho_i)\mathcal{U}(\tilde{\rho}_i^k | \rho_i^{\min}, \rho_i^{\max}), \tag{E.1}$$

Assuming independent observations, the Bayesian estimation for ρ_i on the basis of all

measurements $\tilde{\rho}_i^{r+1}, \ldots, \tilde{\rho}_i^k$ is given by the posterior

$$p(\rho_i, \rho_i | \tilde{\rho}_i^{r+1}, \ldots, \tilde{\rho}_i^k) \propto p_0(\rho_i, \rho_i) \prod_k p(\tilde{\rho}_i^k | \rho_i, \rho_i), \tag{E.2}$$

with $p_0(\rho_i, \rho_i)$ being a prior on the true depth and the ratio of good measurements supporting it. A sequential update is implemented by using the estimation at time step $k-1$ as a prior to combine with the observation at time step k. We refer to [Vogiatzis and Hernández, 2011] for the final formalization and in-depth discussion of the update step.

Note that we consider the depth estimate that is modeled as a Gaussian $\rho_i \sim \mathcal{N}(\hat{\rho}_i, \hat{\sigma}_i^2)$ as converged when its estimated inlier probability $\hat{\rho}_i$ is more than the threshold η_{inlier} and the depth variance $\hat{\sigma}_i^2$ is below σ_{thresh}^2.

Depth Smoothing

The main goal is to filter coarse outliers in the depth estimate but keep the discontinuities in the depth map intact. In [Pizzoli et al., 2014], we utilized a variant of the weighted total variation, introduced by [Bresson et al., 2007] in the context of image segmentation, in order to enforce spatial smoothness of the constructed depth maps. Therefore, we utilize the given depth map $D(\mathbf{u})$ with $\mathbf{u} \in \mathbb{R}^2$ being the image coordinates. For computing the smooth depth map $F(\mathbf{u})$, we apply a variant of the weighted Huber-L1 model as presented in [Pizzoli et al., 2014], that is defined as the variational problem

$$\min_F \int_\Omega \left\{ G(\mathbf{u}) \left\| \nabla F(\mathbf{u}) \right\|_\varepsilon + \lambda(\mathbf{u}) \left\| F(\mathbf{u}) - D(\mathbf{u}) \right\|_1 \right\} d\mathbf{u} \,. \tag{E.3}$$

Note that there are two modifications to the variant presented in [Pizzoli et al., 2014]: (1) first, we use a weighted Huber regularizer that weights the Huber norm according to the image gradient magnitude of the respective reference image by using the weighting function

$$G(\mathbf{u}) = \exp\left(-\alpha \left| \left| \nabla \mathtt{I}_r(\mathbf{u}) \right| \right|_2^q \right). \tag{E.4}$$

This is based on the assumption that image edges of the reference image coincide with depth discontinuities, hence prevents the regularization to smooth across object boundaries. (2) Second, we define $\lambda(\mathbf{u})$, the trade-off between regularization and data fidelity, as a pointwise function depending on the estimated pixel-wise depth uncertainty $\hat{\sigma}^2(\mathbf{u})$ and inlier probability $\hat{\rho}(\mathbf{u})$ of the depth filters:

$$\lambda(\mathbf{u}) = \mathbb{E}[\hat{\rho}(\mathbf{u})] \frac{\sigma_{\max}^2 - \hat{\sigma}^2(\mathbf{u})}{\sigma_{\max}^2}, \tag{E.5}$$

where σ^2_{max} is the maximal uncertainty that the depth filters are initialized with. The confidence value $\lambda(\mathbf{u})$ represents the quality of the convergence of the depth value for each pixel. In the extreme case, if the expected value of the inlier probability $\hat{\rho}(\mathbf{u})$ is zero or the variance is close to σ^2_{max}, the confidence value $\lambda(\mathbf{u})$ becomes zero and solving (E.3) will perform inpainting for these regions.

For solving the optimization problem (E.3), we refer to [Pizzoli et al., 2014] where we defined the primal-dual formulation of the weighted Huber-L1 model. Then, for solving such primal-dual saddle-point problems we utilize the first-order primal-dual algorithm proposed by Chambolle and Pock [Chambolle and Pock, 2011].

Implementation Details

On-board the MAV, only a coarse elevation map is necessary for autonomous maneuvers. This requirement allows us to lower the resolution of the reconstruction, which drastically reduces the processing time for one depth map. In practice, we initialize one depth filter for every 8×8 pixel block in the reference frame. We therefore obtain dense depth maps of size 94×60, totalling 5820 depth filters for every reference image. Given the computing capabilities of our platform, we can update the depth filters in real-time; thus, we do not require to buffer any images and provide frequent updates to the elevation map.

More accurate initialization of the depth filters further reduced the processing time. Hence, we exploit that the visual odometry algorithm already computes a sparse point-cloud of the scene (shown in Figure E.9(b)). We create a two-dimensional KD-Tree of the sparse depth estimates in the reference frame and find for every depth-filter the closest sparse depth estimate. The result is a mosaic of locally-constant depths as shown in the leftmost image of Figure E.5. In case a depth estimate is very close to the depth-filter, the initial depth uncertainty $\hat{\sigma}^2_i$ is additionally reduced for faster convergence. This approach relies on the fact that SVO has few outliers. However, in case of an outlier, we find that the depth-filters do not converge and, thus, no erroneous height measurement will be inserted in the elevation map. In Figure E.5, converged depths are colored in blue and from visual inspection it can be seen that most obvious outliers have not converged. Resulting holes in the elevation map are quickly filled by subsequent updates.

Elevation Map

We make use of a recently developed robot-centric elevation mapping framework proposed in [Fankhauser et al., 2014]. The goal of the original work was to develop a local map representation that serves foot-step planning for walking robots over and

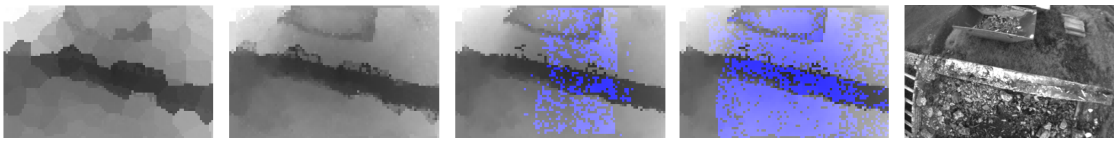


Figure E.5 – Evolution of the depth map reconstruction process. The leftmost image shows the depth map after initialization from the sparse point cloud. After some iterations, the depth filters converge upon which their corresponding pixels are colored in blue. The final depth map is integrated in the elevation map shown in Figure E.8(b). The rightmost image shows the reference image of the depth map.

around obstacles. However, we find that the local two-dimensional elevation map is an efficient on-board map representation for MAVs that are flying outdoors — it allows us to keep a safe distance to the surface and to detect and approach suitable landing spots. By tightly coupling the local map to the robot's pose, the framework can efficiently deal with drift in the pose estimate. The local map has a fixed size, thus, the map can be implemented with a two-dimensional circular buffer that requires constant memory. The two-dimensional buffer allows moving the map efficiently together with the robot without copying any data but by shifting indices and by resetting the values in the regions that move out of the map region. An open-source implementation of the elevation mapping framework is provided by the authors of [Fankhauser et al., 2014].[3]

While the authors of [Fankhauser et al., 2014] used a depth camera, we will demonstrate how the elevation map can be efficiently updated with depth maps computed from aerial monocular views. Furthermore, we extended the framework with a system to switch the map resolution – a requirement that is necessary when the MAV is operating at different altitudes.

Preliminaries

We use three coordinate frames, the inertial world frame W is assumed fix, the map frame M is attached to the elevation map, and C denotes the camera frame attached to the MAV (see Figure E.6). Since the elevation map is robot centric, the translation part of the rigid body transformation $\mathtt{T}_{\mathrm{MC}}(t) = \{\mathtt{R}_{\mathrm{MC}}(t), {}_{\mathrm{M}}\mathbf{t}_{\mathrm{MC}}\} \in SE(3)$ remains fixed at all times. The MAV has an onboard vision-based state estimator that estimates the relative transformation $\mathtt{T}_{\mathrm{WC}}(t) \in SE(3)$.

The elevation map is stored in a two-dimensional grid with a resolution s [m/cell] and width w [m]. The height in each cell (i, j) is modeled as a normal distribution with mean $\hat{h}$ and variance $\hat{\sigma}_h^2$.

[3] Available at http://github.com/ethz-asl/grid_map

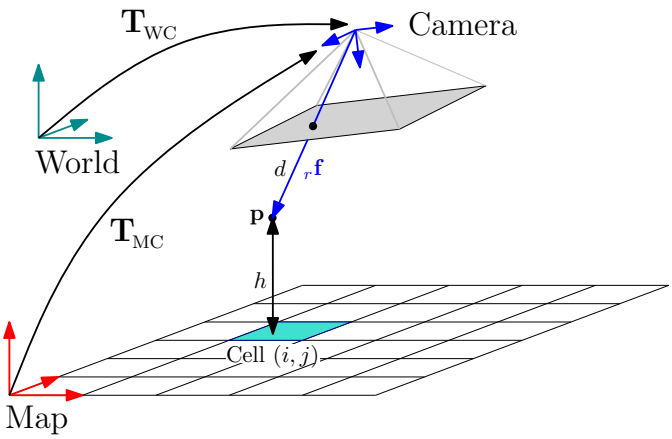


Figure E.6 – Notation and coordinate frames used for the elevation map.

Map Update

In Section E.3, we described how to process data from a single camera to obtain a probabilistic depth estimate $d \sim \mathcal{N}(\hat{\rho}, \hat{\sigma}_d^2)$ corresponding to a pixel $\mathbf{u}$ in a selected reference image $\mathtt{I}_r$, where $r = \mathrm{C}(t_r)$ denotes the camera frame C at a selected time instant t_r. Given the probabilistic depth estimate, we can follow the derivation in [Fankhauser et al., 2014] to integrate a measurement in the elevation map. In the following, we summarize the required steps.

Given the depth estimate $\hat{\rho}$ of pixel $\mathbf{u}$, we can find the corresponding 3D point $\boldsymbol{\rho}$ by applying the camera model:

$$ {}_{\mathrm{C}}\bar{\boldsymbol{\rho}} = \pi^{-1}(\mathbf{u}) \cdot \hat{\rho}, \tag{E.6} $$

where π^{-1} is the inverse camera projection model that can be obtained through camera calibration. The prescript c denotes that the point ${}_c\bar{\boldsymbol{\rho}}$ is expressed in the camera frame of reference and the bar indicates that the point is expressed in homogeneous coordinates. We find the height measurement by transforming the point to map coordinates and applying the projection matrix $\mathbf{P} = [0\ 0\ 1]$ that maps a 3D point to a scalar value:

$$ \tilde{h} = \mathbf{P}\ \mathtt{T}_{\mathrm{MC}}\ {}_{\mathrm{C}}\bar{\boldsymbol{\rho}}. \tag{E.7} $$

To obtain the variance of the measurement, we need to compute the Jacobian of the projection function (E.7):

$$ \mathbf{J}_{\mathbf{P}} = \frac{\partial \tilde{h}}{\partial_{\mathrm{C}}\boldsymbol{\rho}} = \mathbf{P}\ \mathbf{R}_{\mathrm{MC}}, \tag{E.8} $$

where $\mathbf{R}_{\mathrm{MC}} \in SO(3)$ is the rotational part of $\mathtt{T}_{\mathrm{MC}}$. The variance of the measurement can

then be written as:

$$\tilde{\sigma}_h^2 = \mathbf{J_P} \mathbf{\Sigma}_\rho \mathbf{J_P}^\top. \tag{E.9}$$

Note that Equation (E.9) can be extended with the uncertainty corresponding to the robot pose T_{CW} as derived in [Fankhauser et al., 2014]. The uncertainty of the 3D point $\mathbf{\Sigma}_\rho$ is derived as follows:

$$\mathbf{\Sigma}_\rho = \mathbf{R}\ \mathrm{diag}\left(\frac{\hat{\rho}}{f}\sigma_p^2, \frac{\hat{\rho}}{f}\sigma_p^2, \sigma_d^2\right)\ \mathbf{R}^\top, \tag{E.10}$$

where $\mathbf{R}$ is a rotation matrix that aligns the pixel bearing vector $\mathbf{f}$ with the z-axis of the camera coordinate frame C. The fraction $\frac{\hat{\rho}}{f}$ projects the pixel uncertainty σ_p^2 (set fixed to one pixel) to the 3D space, using the focal length f of the camera.

Given the height measurement mean $\tilde{h}$ and variance $\tilde{\sigma}_h^2$, we can update the height estimate in the corresponding cell (i, j) using a recursive Bayesian update step:

$$\hat{h} \leftarrow \frac{\tilde{\sigma}_h^2 \hat{h} + \hat{\sigma}_h^2 \tilde{h}}{\tilde{\sigma}_h^2 + \hat{\sigma}_h^2}, \qquad \hat{\sigma}_h^2 \leftarrow \frac{\tilde{\sigma}_h^2 \hat{\sigma}_h^2}{\tilde{\sigma}_h^2 + \hat{\sigma}_h^2}. \tag{E.11}$$

Map-Resolution Switching

Given the height z of the robot above ground in meters, the focal length f in pixels, and the fact that we initialize a depth-filter for every image block of 8 pixels size, we can compute the optimal elevation map resolution:

$$s_{\mathrm{opt}} = \frac{8}{f} \cdot z \qquad [\mathrm{m/cell}]. \tag{E.12}$$

For instance, when flying at a height of 5 meters with our camera that has a focal length $f = 420$ pixels, the optimal resolution would be 0.1 meters per cell. We limit the size of the map to have 100 by 100 cells; thus, depending on the resolution, a larger or smaller area is covered by the elevation map.

During operation, we maintain an estimate of the optimal resolution s_{opt} and compare it with the currently-set resolution s_{cur}. If the MAV is ascending and the optimal resolution increases by a factor $\alpha_{\mathrm{up}} = 1.8$ compared to the current resolution, we double the resolution. Similarly, if the optimal resolution reduces by a factor of $\alpha_{\mathrm{down}} = 0.6$, i.e., the MAV is approaching the surface, we reduce the resolution by half. Additionally, we limit the minimal resolution to 5 cm per cell to avoid changing the resolution too often during the landing procedure. When the resolution changes, we down or up-sample all values in the map using bilinear interpolation.

Landing Spot Detection

To motivate the usefulness of the proposed local elevation map, we implemented a basic landing-spot detection and landing system that is described in the following.

Let us define a 3D point $\boldsymbol{\rho}$ located on the surface of the terrain and within the range of the local elevation map. The point has discrete coordinates (i, j) in the two-dimensional elevation map and is located at height $h(i, j)$.

We define a safe landing spot to have a local neighborhood of radius r in which the surface is flat. The radius r is related to the size of the MAV. We formalize this criterion with the cost function:

$$C(i,j) = \sum_{(u,v)\in\mathcal{R}(i,j,r)} ||h(u,v) - h(i,j)||^2, \tag{E.13}$$

where $\mathcal{R}(i, j, r)$ is the set of cells around coordinate (i, j) that are located within a radius r.

Experimentally, we find a threshold $C_{\max}$ that defines the acceptable cost to be a safe landing spot. We compute a binary mask of all cells in the elevation mask, which have a cost lower than $C_{\max}$. Subsequently, we apply the distance transform to the binary mask in order to find the coordinates (i, j) that have a cost lower than the threshold and are farthest from all cells that do not satisfy the criterion. Thus, the final landing spot should be as far as possible from any obstacles.

From the construction of the elevation map, it may well be that a cell does not have an elevation value. This is the case in regions that have not been measured before or in which the depth filters did not converge, e.g., due to lack of texture, reflections, or due to moving objects. Therefore, before applying the kernel in Equation (E.13) to the elevation map, we set all cells without an elevation value to $C_{\max}$. Thereby, assuring to land in regions where depth computation is feasible, thus, landing is more likely to be safe.

Experiments

We performed experiments of the elevation-mapping and landing system both indoors in a quadrotor testbed as well as outdoors. Videos of the experiments can be viewed at: http://rpg.ifi.uzh.ch

The quadrotor used for all experiments is shown in Figure E.7. It is equipped with a MatrixVision mvBlueFOX-MLC200w 752 × 480-pixel monochrome global shutter camera, an 1.7 GHz quad-core smartphone processor running Ubuntu, and an PX4FMU

Figure E.7 – Experimental platform with (1) down-looking camera, (2) on-board computer, and (3) inertial measurement unit.

autopilot board from Pixhawk that houses an Inertial Measurement Unit (IMU). In total the quadrotor weights less than 450 grams and has a frame diameter of 35 cm. More details about the experimental platform are given in [Faessler et al., 2015].

Timing Measurements

All processing during the experiments was done on the on-board computer using the ROS[4] middleware. During operation, the elevation-mapping and landing module uses, on average, one processing core, SVO and MSF together another two cores, and the fourth core is reserved for the camera driver, communication, and control.

Table E.1 lists the timing measurements. On average, the depth map requires 6 to 10 images for convergence. However, this depends greatly on the motion of the camera as well as the depth and the texture of the scene [Forster et al., 2014a]. For the listed measurements, we were flying at a speed of approximately 1.5 m/s and at a height of 1.8 meters. Updating all depth estimates with a new image requires on average 150 milliseconds. Once 50% of the depth filters in the depth map have converged, we filter the resultand depth map by solving the gradient-weighted Huber-L1 model (E.3) (130 milliseconds) and integrate the smoothed depth map in the elevation map (10 milliseconds).

Summarizing, the mapping module receives images from the camera at 10 Hz and integrates approximately 6-10 images to output one depth map per second.

Once it is necessary to detect a landing spot, it requires approximately 268 milliseconds to compute the landing cost (E.13) for all 10,000 grid cells and to find the best landing-spot in the current elevation map.

[4] http://www.ros.org

	Mean [ms]	Median [ms]	Std. Dev. [ms]
Depthmap update	150	143	40
Regularization (10 iterations)	133	129	19
Elevation map update	10	10	2
Total time for one depthmap:	1098	999	503
Landing spot detection	268	268	19

Table E.1 – On-board timing measurements.

Outdoor mapping experiment

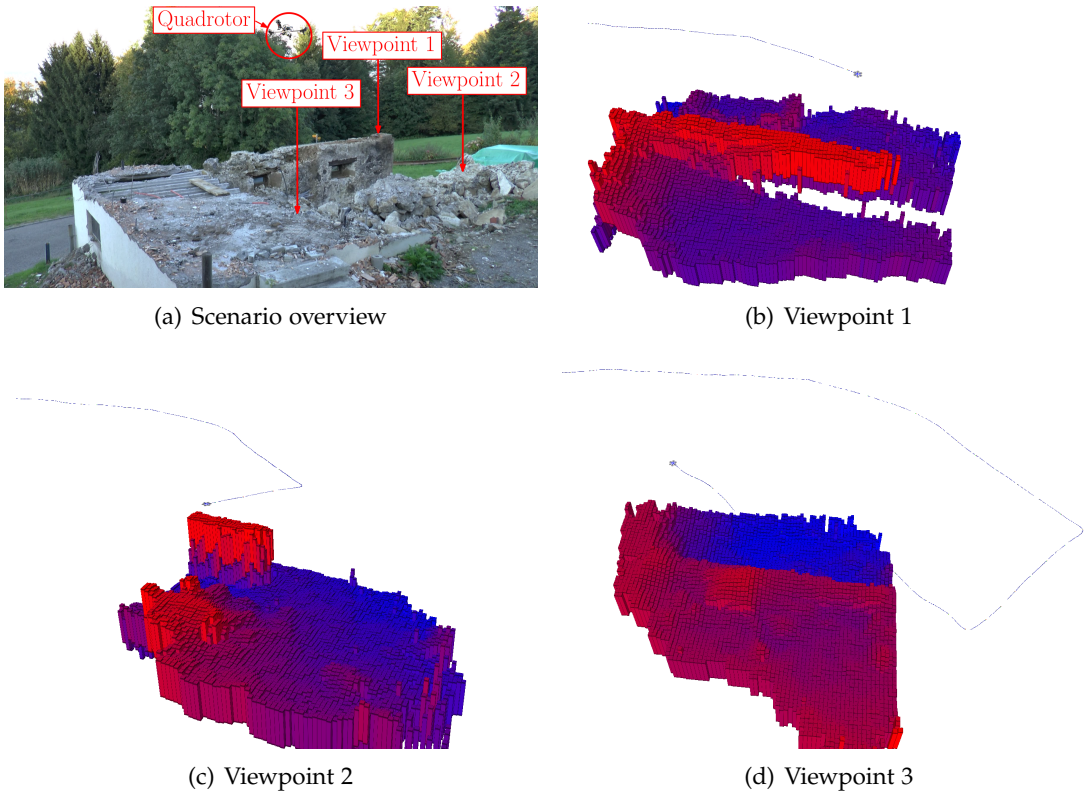
For the outdoor elevation-mapping experiment, the quadrotor was commanded by a remote operator under assistance of the on-board vision-based controller, i.e., the operator could command the quadrotor directly in x-y-z-yaw space. On average, the quadrotor was flying 4-5 meters above the surface and used an elevation map of size 10 by 10 meters with a resolution of 10 centimeters per cell. The terrain consisted of a teared-down house, rubble, asphalt, and grass. Figure E.8(a) gives an overview of the scenario and indicates the location of the MAV for the elevation maps shown in Figures E.8(b) to E.8(d). The complete mapping process can be viewed in the video attachment of this work.

An update rate of 1 Hz is sufficient to always maintain a dense elevation map below the MAV. However, when moving in a straight line, the local elevation map behind the MAV is more populated than in the front. In the future, we will modify the MAV to have a slightly forward facing camera in order to have a more even distribution.

The system can cope with drastic elevation changes as well as challenging surfaces such as grass and asphalt that are characterized by high-frequency texture. Due to the probabilistic approach to depth estimation, which uses multiple measurements until convergence, we observe very few outliers in the elevation map. Note that we only insert depth estimates in the elevation map that have actually converged. In untextured regions, paths with of dynamic motion or zones with reflecting surfaces, such as water, the depth filter does not converge and, thus, the elevation map remains empty. As visible in Figure E.8(b), the elevation map remains also empty in occluded areas.

Landing experiment

We performed autonomous landing experiments both indoors and outdoors as demonstrated in the supplemental video. Figure E.9(a) shows the indoor testbed that contains textured boxes as artificial obstacles. The elevation map after a short exploration is



(a) Scenario overview (b) Viewpoint 1

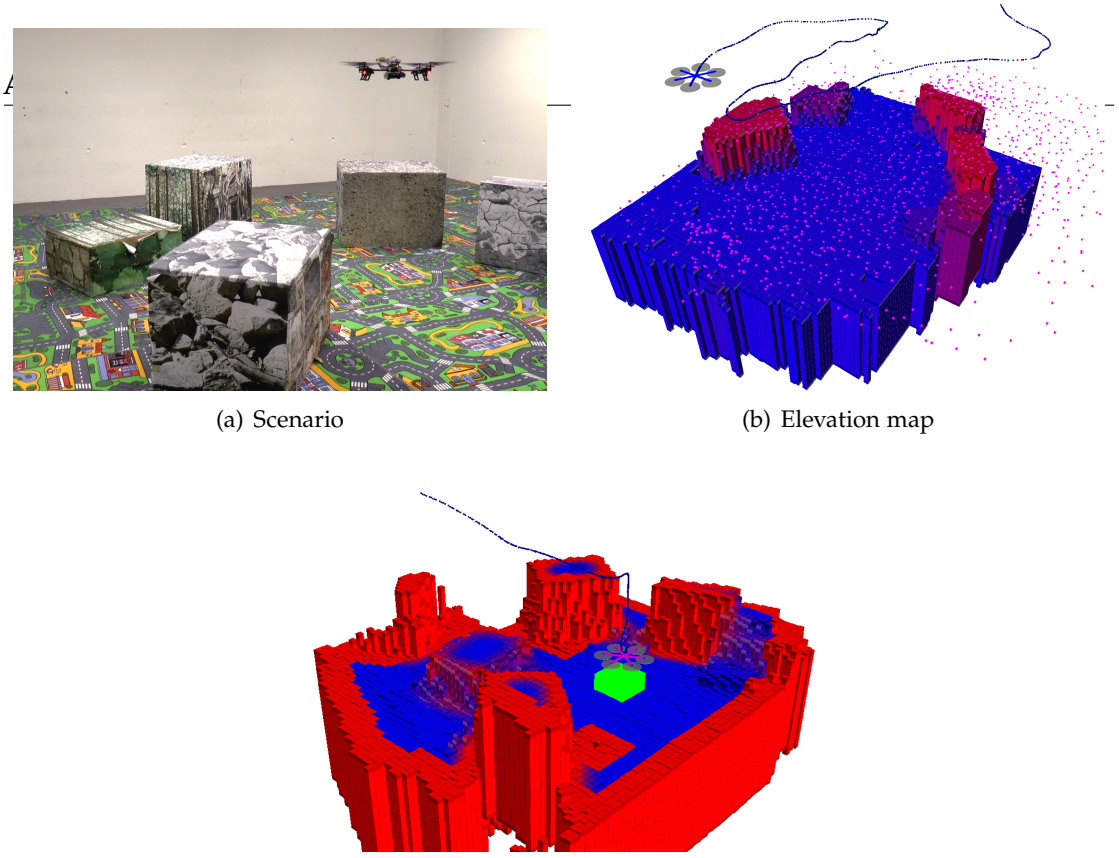
(c) Viewpoint 2 (d) Viewpoint 3

Figure E.8 – Excerpts from the video attachment. The quadrotor is flying over a destroyed building. Figures E.8(b) to E.8(d) show the elevation map at three different times. The corresponding viewpoints are marked in the scenario overview in Figure E.8(a). Note that the elevation map is local and of fixed size. Its center lies always below the quadrotor's current position.

displayed in Figure E.9(b). Once the MAV receives a command to land autonomously, it computes the landing score for the current elevation map and selects the best spot as described in Section E.5. In Figure E.9(c), the elevation map is colored with the landing cost that is formalized in Equation (E.13). Blue means that the area is flat and, thus, safe for landing. The algorithm selects the point that is farthest from any dangerous area (colored red) and marks it with a green cube. The MAV then autonomously approaches a way-point vertically above the detected landing spot and then slowly descends until vision-based tracking is lost, which is typically at less than 30 cm above ground. Subsequently, the MAV continues blindly to descend until impact is detected and the motors turn off.

Conclusion

In this paper, we proposed a system for mapping the local ground environment underneath an MAV using only a single camera and on-board processing resources.



(a) Scenario

(b) Elevation map

(c) Landing procedure

Figure E.9 – Excerpts from the video attachment. Figure E.9(a) shows the indoor flying arena with textured obstacles. The MAV first explores the arena and creates an elevation map of the surface that is shown in Figure E.9(b). The pink points illustrate the sparse map built by the on-board visual odometry system and which are used to initialize dense depth estimation. Figure E.9(c) shows the detected landing spot that is marked as green cube and the MAV that is shortly before impact. The blue line is the trajectory that the MAV flew to approach the landing spot.

We advocate the use of a local, robot-centric, and two-dimensional elevation map since it is efficient to compute on-board, ideal to accumulate measurements from different observations, and is less affected by drifting pose estimates. The elevation map is updated at a rate of 1 Hz with probabilistic depth maps computed from multiple monocular views. The probabilistic approach results in precise elevation estimates with very few outliers even for challenging surfaces with high frequency texture, e.g., asphalt. To highlight the usefulness of the proposed mapping system, we successfully demonstrated autonomous landing-spot detection and landing.

F Air-Ground Localization Using Dense Reconstruction

C. Forster, M. Pizzoli, and D. Scaramuzza. Air-ground localization and map augmentation using monocular dense reconstruction. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3971–3978, 2013. URL http://dx.doi.org/10.1109/IROS.2013.6696924.

Air-Ground Localization and Map Augmentation Using Monocular Dense Reconstruction

Christian Forster, Matia Pizzoli, Davide Scaramuzza

Abstract — We propose a new method for the localization of a Micro Aerial Vehicle (MAV) with respect to a ground robot. We solve the problem of registering the 3D maps computed by the robots using different sensors: a dense 3D reconstruction from the MAV monocular camera is aligned with the map computed from the depth sensor on the ground robot. Once aligned, the dense reconstruction from the MAV is used to augment the map computed by the ground robot, by extending it with the information conveyed by the aerial views. The overall approach is novel, as it builds on recent developments in live dense reconstruction from moving cameras to address the problem of air-ground localization. The core of our contribution is constituted by a novel algorithm integrating dense reconstructions from monocular views, Monte Carlo localization, and an iterative pose refinement. In spite of the radically different vantage points from which the maps are acquired, the proposed method achieves high accuracy whereas appearance-based, state-of-the-art approaches fail. Experimental validation in indoor and outdoor scenarios reported an accuracy in position estimation of 0.08 meters and real time performance. This demonstrates that our new approach effectively overcomes the limitations imposed by the difference in sensors and vantage points that negatively affect previous techniques relying on matching visual features.

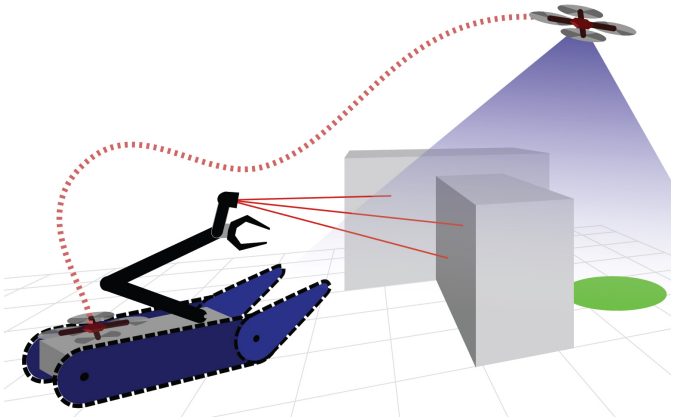


Figure F.1 – The flying robot operating in close range to the ground robot provides a different vantage point for human tele-operators in a search-and-rescue scenario. We address the problem of autonomously localizing the aerial robot with respect to the ground-robot based on the structure of the scene.

Introduction

A heterogeneous robotic system consistent of both, ground and aerial robots of different sizes, shapes and with different sense-act capabilities could greatly assist professional rescuers in a search and rescue scenario. However, it is difficult for the same human operator to concurrently monitor and navigate multiple robots while coordinating with other operators. Therefore, the necessary technologies must be developed to allow heterogeneous robots to autonomously localize and move with respect to each other and thereby ease the task of the operator and provide the best possible situation awareness.

In this work we consider a single MAV that acts as a "flying external eye" for a ground robot. The MAV operates in close range to the ground robot and offers the ability to hover and move in complex three dimensional space and observe the scene from a vantage point inaccessible to the ground robot (see Figure F.1). The use of very small and lightweight MAVs reduces safety concerns, costs, and increases the agility of the platform. However, active ranging devices such as laser rangefinders or RGBD sensors cannot currently be used due to payload and power consumption restrictions. The ground robot, on the other hand, can carry more payload such as active depth sensors, processors and may be equipped with a manipulator arm. The usefulness of such a heterogeneous robot team in a disaster scenario has recently been demonstrated in [Michael et al., 2012].

In this paper, we address the problem of localizing the MAV with respect to the ground robot in close range. This capability will allow the robots to execute collaborative tasks and to present the teleoperator with a ground map which is augmented with aerial

views from the MAV.

Due to payload restrictions, the MAV is equipped with a single downward-looking camera. On the other hand, the ground robot has a range sensor (either a laser or an RGBD camera) and further carries the main processing unit. Our experimental platforms are depicted in Figure F.13.

Given the available sensory capabilities, there are two possible strategies to mutually localize the robots: (i) by leveraging relative observations between the MAV and the ground robot [Rudol et al., 2008], (ii) or by matching and aligning maps computed by the MAV and the ground robot. The second option offers the advantage that the robots do not need to remain in the field of view of each other. However, the main challenge in the second strategy is the drastically different view points of the two robots (see Figure F.2).

In this paper we propose a novel solution to this problem by leveraging the 3D surface computed from different view points and heterogeneous sensors. Through the alignment of both maps, the relative pose of the robots can be recovered. Computing a dense 3D surface from monocular cameras in real-time has only recently become feasible with the use of GPGPU computing [Newcombe et al., 2011b, Rhemann et al., 2011]. Therefore, we propose to distribute the processing between the robots. The MAV computes its relative motion onboard using the downward looking camera [Weiss et al., 2011b] and, additionally, streams the video to the ground robot where a dense 3D model is computed and aligned with the ground robot's 3D map. Thereby, the relative pose of the robots is recovered.

We propose a solution for global alignment of the aerial and ground maps based on Monte Carlo Localization on the height-maps. Subsequently, the estimated transformation is refined through an Iterative Closest Point (ICP) algorithm. In experimental results we show that in a cluttered environment with sufficient 3D structure, we can compute the relative position with a precision of 0.078 m. Furthermore, we illustrate how the ground-based map can be augmented with the aerial view.

The outline of the paper is the following. In Section F.2 we compare our approach to related works in the literature, while Section F.3 provides an overview of the proposed method. In Section F.4 we present the SLAM methods operating on the MAV and the ground robot, while in Section F.5 the dense reconstruction method is detailed. In Section F.6, our Monte Carlo approach to global localization is described and in Section F.7 we present the iterative pose refinement. Section F.8 reports about the experimental validation and in Section F.9 the conclusion is drawn.

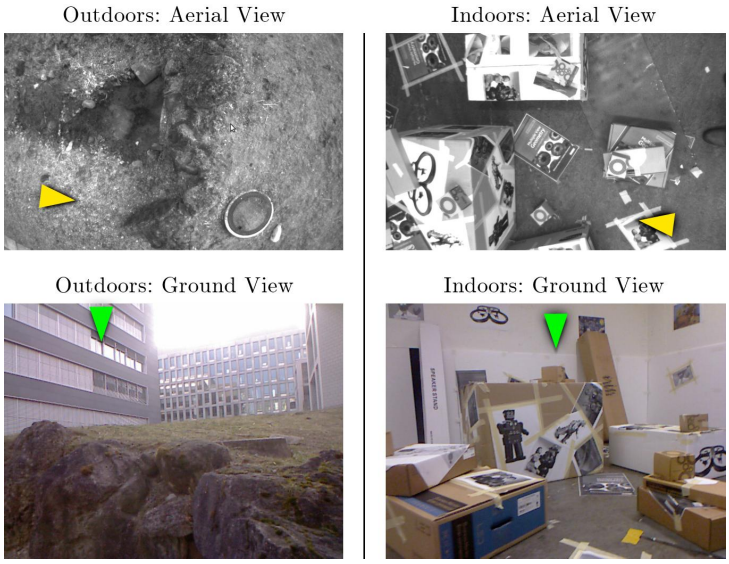


Figure F.2 – Outdoor (left) and indoor (right) scenes observed from aerial and ground point of views. Robot poses are expressed by the arrows: yellow for the ground robot and green for the MAV.

Figure F.3 – Image feature matching results of the indoor scene using ASIFT [Morel and Yu, 2009]. Matches were found on planar textured surfaces. No matches were found in the outdoor scene depicted on the left of Figure F.2.

Related Work

Very little research has addressed close-range relative localization of aerial and ground-robots. In most related works, the aerial robot flies outdoors higher than 20 meters, and can be localized using GPS [Hsieh et al., 2007, Stentz et al., 2002, Vidal-Calleja et al., 2011]. To the best of our knowledge, the work in [Michael et al., 2012] is the first to demonstrate how a MAV could assist a ground robot in close collaboration in mapping a damaged building indoors. In their configuration, the ground-robot is equipped with a laser rangefinder and the flying robot with both a laser rangefinder and a RGBD sensor. The computed maps are aligned under the assumption that the ground robot does not move during the flight of the MAV. It is not mentioned whether the global map computation is executed onboard or in an offline stage and

no processing times are reported. In our work, we investigate the relative localization, which is a precondition to the mapping task. Furthermore, we do not require that the ground-robot remains still while the flying robot is mapping and provide continuous relative position information in real-time.

Photorealistic modeling of urban scenes addresses a registration problem related to ours [Ding et al., 2008, Liu and Stamos, 2006]. Similar to our work, the one described in [Zhao et al., 2005] addresses the computation of a 3D point-cloud from aerial video using dense motion stereo and the alignment with a ground-based map. Wendel [Wendel et al., 2011] proposes to align a 3D reconstruction created by a MAV with a Digital Surface Model (DSM), where an initial alignment is provided by GPS information and a refined alignment is computed by evaluating the correlation between a height map computed from the reconstruction and the DSM. The time for alignment takes about 10 minutes, depending on the number of points and resolution. Our work advances the state of the art in two important aspects: (i) dense monocular maps are effectively used for MAV localisation and (ii) the integrated system can operate in real time, which is a desirable feature in most robotic perception tasks.

Registration methods based on image appearance require finding matches between the visual features in the aerial and ground views. Recently, advancements have been made in the field of wide-baseline image matching [Morel and Yu, 2009, Donoser and Bischof, 2006, Wu et al., 2008]. Many state-of-the-art approaches are grounded on the method described in [Morel and Yu, 2009] and aim at providing affine invariance by computing feature descriptors after a set of pre-defined warping transformations have been applied to the images to be matched. These methods implicitly rely on a piecewise planarity assumption, which is satisfied in many man made scenarios but does not hold in general. An example is provided in Figure F.2. The aerial and ground views are shown from our validation dataset in case of indoor and outdoor scenarios. Figure F.3 displays the results for a feature matching algorithm based on the work in [Morel and Yu, 2009] on the images corresponding to the ground and aerial views. Noticeably, the method managed to find few correct matchings on the planar box surfaces. The same method, applied to the views in the left column of Figure F.2, returned no correct matches. The required processing time (about 6 seconds for feature extraction and 27 for matching) constitutes another important limitation to the application of robust approaches for visual feature matching to the problem of real time localisation.

To overcome these limitations, instead of relying on visual feature matching between the views from the MAV and the ground robot, our new approach exploits the 3D structure, which is computed by the MAV through monocular dense reconstruction and by the ground robot making use of its range sensor. This approach is novel and constitutes the actual contribution of this paper.

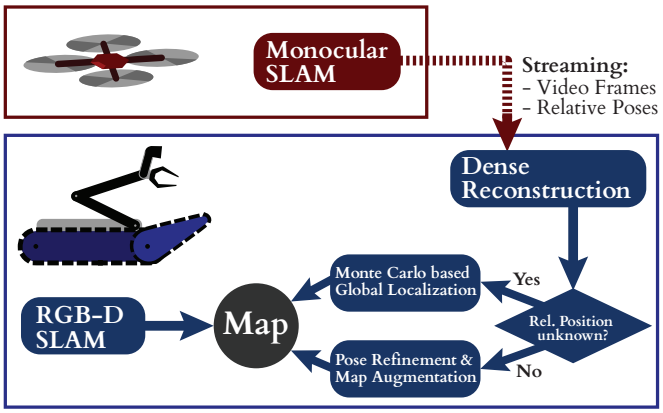


Figure F.4 – Illustration of the localization and mapping pipeline. Each building block is described in detail in Sections F.4 to F.7.

System Overview

Figure F.4 provides an overview of the proposed system. The MAV is equipped with a single downward-looking camera and an IMU. A monocular SLAM algorithm runs onboard the MAV to estimate its egomotion. The absolute scale is recovered by a Kalman Filter [Nuetzi et al., 2011]. The MAV streams the video to the ground robot together with relative-pose estimates for every frame.

On the ground robot, a set of subsequent frames received from the MAV are used to compute a *dense* map through leveraging all information in the monocular images—not only salient corner points. Real-time performance is achieved through a highly parallelized GPU implementation. The ground-robot is further equipped with a Kinect sensor to create a second—ground-based—3D map by means of an RGBD SLAM system.

For the alignment of the aerial and ground maps, we propose two solutions: If an *a priori* guess is available for the relative pose of the two robots, their maps are aligned using ICP. Otherwise, we propose a *Monte Carlo Localization* (MCL) based method to globally localize the MAV with respect to the ground robot. The MCL method relies on correlating the height-maps computed from the two vantage points.

The proposed pipeline requires an overlap between the aerial and ground maps and a 3D structure in the scene. In a completely flat environment, the algorithm does not converge. Hence, the proposed method is a strong complement to image feature based methods which fail to match in cluttered environments at such radically diffent viewpoints.

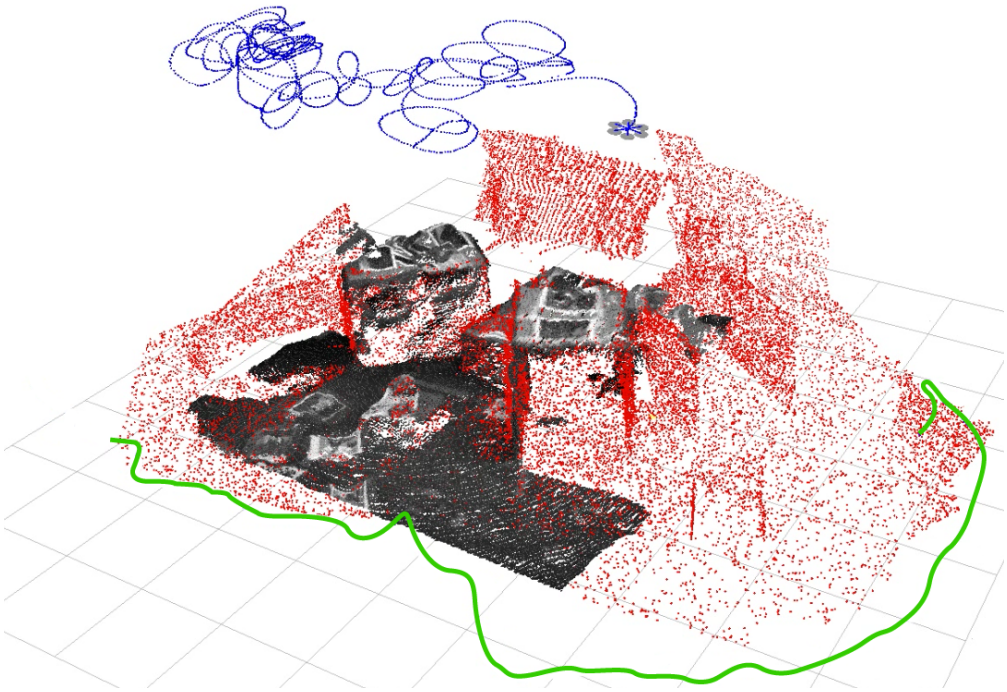


Figure F.5 – Air-ground localization and map augmentation.The trajectories of the aerial and ground robots are displayed in blue and green respectively. The map computed by the ground robot (displayed in red) is augmented with the dense reconstruction from the aerial views (displayed in greyscale).

SLAM on the MAV and the Ground Robot

The SLAM system on the flying robot implements the keyframe-based monocular *Visual Odometry* (VO) pipeline by Kneip et. al [Kneip et al., 2011a]. It is boosted in terms of robustness and efficiency by including incremental relative rotation priors obtained from the onboard IMU.

On the ground robot, an RGBD sensor is used to create the map. Our RGBD SLAM system is a modification of the monocular SLAM algorithm described above. Notably, we are able to speed-up triangulation using the depth provided by the sensor as a prior. Additionally, the depth measurements are used to initialize map-points in case of pure rotation of the camera.

Both SLAM systems could also be replaced with state-of-the-art algorithms such as [Klein and Murray, 2007, Strasdat et al., 2011, Engelhard et al., 2011, Pomerleau et al., 2011].

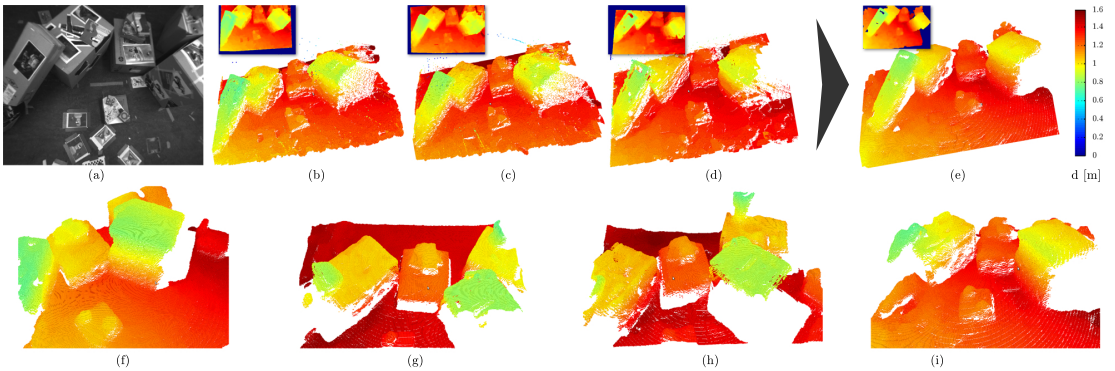


Figure F.6 – Point clouds computed by monocular dense reconstruction for one indoor evaluation dataset. Depth maps are also shown in insets. In (a) the reference view is shown. Figures (b)-(d) correspond to different depth computations, fused into the depth map of Figure (e) through the algorithm presented in Section F.5. Figures (f)-(i) show several results from the fusion algorithm computed as the MAV flies over the experimental scenario. The color code refers to the distance d from the camera acquiring the reference view.

Dense Monocular Reconstruction

In this section we describe a method to estimate the dense point cloud from the images collected by the MAV as it flies over an area of interest. Timestamped views and camera poses are streamed to the ground robot, where the computation can take advantage of the multi-cores architecture offered by the onboard GPU, an Nvidia Quadro K2000M in our experiments.

The solution we propose to estimate a dense point cloud from monocular views with known camera motion derives from *Multi View Stereo* methods [Furukawa and Ponce, 2010] and is motivated by the following facts: i) assuming constant brightness, frames taken from close viewpoints allow high quality matching; ii) a large baseline among views enables a more reliable depth estimation and outlier rejection. Therefore, similarly to [Newcombe et al., 2011b], we propose to compute depth maps from a large number of highly overlapping views, yielding a coarse, but very fast estimation. Filtering and regularization have been proposed to improve the accuracy of the depth maps computed from aggregation of the photometric error in stereo [Rhemann et al., 2011, Newcombe et al., 2011b]. Being computed from close views, these depth maps may still contain wrong estimations. For this reason, we chose to integrate several depth maps, which are computed as the MAV flies over the area of interest. This is due to the fact that—differently from those previous works mainly concerned with the recovery of visually appealing reconstructions—we are interested in accurate maps, useful for localization. Thus, we aim at rejecting wrong estimations that would negatively affect the alignment performance. Further, the use of the recovered structure for the air-ground localization imposes severe constraints in computing time (cfr. Table F.1 for average computing times). We chose to rely on the range-image fusion algorithm

introduced in [Zach, 2008]. The algorithm is robust against wrong estimations; it is reported to be accurate and it is highly parallelizable, which makes it suitable for computation on a modern graphics card.
The depth maps are converted into distance fields $f_i : \Omega \to [-1, 1]$ defined on a voxel space $\Omega \subset \mathbb{R}^3$ specifying the volume of interest, and compressed into a histogram representation to reduce the memory footprint. At every voxel v, the values of f_i encodes the distance to the surface according to the i-th depth map and is approximated by evenly-spaced bin centers c_j.
Let $n(v, j)$ denote the histogram count of bin j at voxel v. The depth map fusion consists in estimating the distance field ϕ given the hypotheses represented by f_i and is computed by the following minimization of an energy functional:

$$\min_{\phi} \int_{\Omega} \left\{ |\nabla \phi| + \lambda \sum_{j} n(v, j) |\phi(v) - c_j| \right\} dv. \tag{F.1}$$

The data term $\sum_j n(v, j)|\phi(v) - c_j|$ approximates the distance of the solution from the distance fields, while the regularization term $|\nabla \phi|$ penalizes the surface area, removing errors due to outliers and approximating the surface in case of missing depth data. λ is a tunable parameter weighting the data term. The minimization in Equation F.1 follows an iterative approach based on gradient descent.

The integrated surface is implicitly defined by the zero level set of the ϕ function and a point cloud is then computed through ray casting (see, for example, [Izadi et al., 2011] Listing 3). Figure F.6 depicts the process of fusing 3 dense structure estimations by the MAV (subfigures (b)-(d)) into one regularized structure (subfigure (e)). The algorithm effectively rejects erroneous estimations that are not supported by the majority of the depth maps. Different examples of dense reconstructions from the MAV views are reported in the second row of Figure F.6. Once computed, the structure is made available for localization, as explained in the following sections.

Global Localization

In this section, we describe a method to globally localize the MAV with respect to the ground robot based on 3D point-clouds computed from the two perspectives. Since the MAV operates in close range to the ground robot, we assume the global search region to be approximately 3m around the ground robot position.

The standard method to align two image-based maps is to find corresponding features (points, lines, edges, planes) either in the 2D images or the 3D pointcloud [Leung et al., 2008, Rusu and Cousins, 2011]. However, we want to make no assumption on any regularity in the scene such as piecewise planarity. Additionally, both the aerial and ground based map can contain missing data and may not be fully overlapping.

We solve the problem through searching for the highest correlation between height-maps computed from the two pointclouds. In order to deal with local minima of the alignment, the procedure is extended with a Monte Carlo Localization method that verifies many hypotheses over the course of multiple pointclouds computed by the MAV. This extension is described thereafter.

Height-Map Alignment

In our setting, both the MAV and ground robot are equipped with an IMU. This provides the gravity vector, which can be used to project their maps to the ground plane (see Figure F.7). This results in the height maps that we denote with H_a and H_g respectively. The height maps are defined on discrete 2D grids Ω_a and Ω_g with a resolution of 50 cells per meter. If multiple points project on to the same grid cell, the highest point is selected. Furthermore, we apply a morphological dilation operator on the height-maps of 3×3 cells in order to fill holes. Holes denote cells of the height-map with missing height data.

The best alignment of the two height maps in the predefined search region is found by searching for the relative pose $\mathbf{u}$ with the minimum *Zero Mean Sum of Squared Differences* (ZMSSD) of the two maps:

$$C(\mathbf{u}) = \eta \sum_{\mathbf{x}\in\bar{\Omega}(\mathbf{u})} \Big\{ [H_a(\mathbf{x}) - \hat{H}_a] - [H_g(\mathbf{x}+\mathbf{u}) - \hat{H}_g(\mathbf{u})] \Big\}^2, \tag{F.2}$$

where $\hat{H}_g(\mathbf{u})$ and $\hat{H}_a$ are the mean of the height maps in the overlapping area at the relative position $\mathbf{u}$ and $\eta = 1/(2|\bar{\Omega}(\mathbf{u})|)$ is a normalization factor. Furthermore, $|\bar{\Omega}(\mathbf{u})| = |\Omega_a(\mathbf{u}) \cap \Omega_g|$ denotes the number of cells on which a height is defined for both, the translated aerial height-map $H_a(\mathbf{u})$ and the ground-based height-map H_g. The final relative position $\tilde{\mathbf{u}}$ corresponds to the minimum ZMSSD:

$$\tilde{\mathbf{u}} = \arg\min_{\mathbf{u}} \; C(\mathbf{u}) \tag{F.3}$$

The advantage of the ZMSSD cost is that the local normalization makes the alignment independent of the z location, whereas the final alignment in the vertical z axis can easily be recovered with $\Delta z = \hat{H}_g(\tilde{\mathbf{u}}) - \hat{H}_a$. However, the ZMSSD cost does not equalize the average relative heights between the two height-maps which is in contrast to the correlation cost which is applied in [Wendel et al., 2011].

The search is done over $\mathbf{u} = [x, y] \in \mathbb{R}^2$ since in the experiments the magnetic north direction could be recovered from the IMU. Depending on the environment, this measurement can be less reliable which would require to add the heading direction to $\mathbf{u}$. This extension is straightforward but has the drawback that the computation

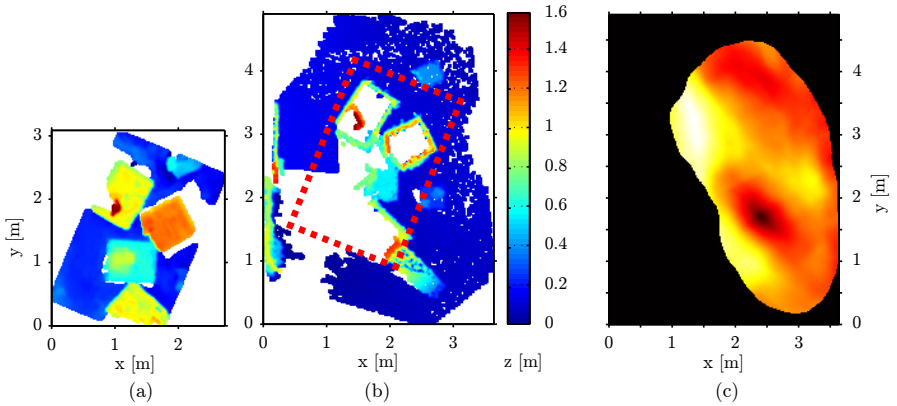


Figure F.7 – The height maps of an aerial and a ground-based map are illustrated in Figures (a) and (b) respectively. The red dotted square in (b) shows the best alignment of the two height-maps. The costmap in (c) illustrates the ZMSSD cost for every possible relative position $\mathbf{u}$ of the two maps. A global minima is located at the dark spot.

time increases exponentially and there can be more local minima in the cost which can however be recovered with the filter described in the next section.

Furthermore, the search space is limited by a minimum overlap between the two height-maps $|\bar{\Omega}(\mathbf{u})| > \theta_{\text{overlap}}$. We set the threshold to 50% of the area of the aerial height-map $|\Omega_a|$. This is the reason for the curvy boarder in the costmap illustrated in Figure F.7c.

Monte Carlo-based Alignment

Due to self-similarities in the environment, the costmap computed in the previous section may contain multiple local minima. We propose to apply Monte Carlo Localization with mixture proposal distribution [Thrun et al., 2005, p. 262]. This allows us to track multiple hypotheses over the course of several height-maps computed from the MAV in order to identify the true relative position. We represent the belief of the relative position with a set $\mathcal{U}$ of M particles:

$$\mathcal{U} = \mathbf{u}^{[1]}, \mathbf{u}^{[2]}, \ldots, \mathbf{u}^{[M]}. \tag{F.4}$$

For the first height-map from the MAV the cost for each relative position is computed within the search region which results in the costmap of Figure F.7c. M particles are then sampled from the costmap with probability

$$p(\mathbf{u}) \sim \exp\left\{-\frac{C(\mathbf{u})}{\sigma}\right\}, \tag{F.5}$$

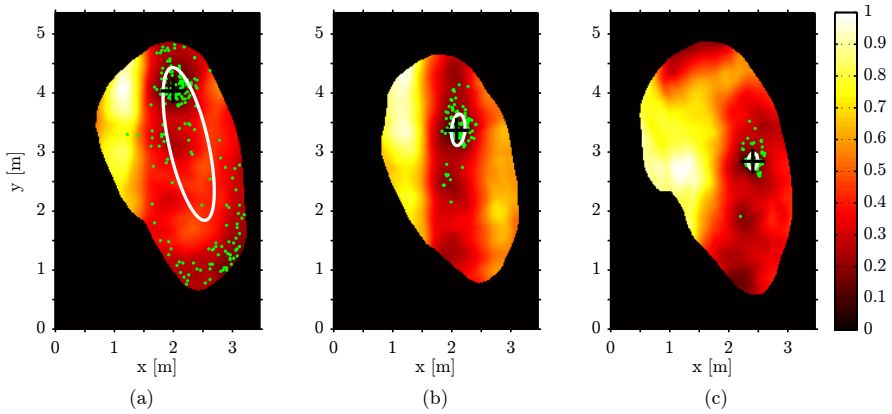


Figure F.8 – Evolution of the particles in the Monte-Carlo–based global alignment. The costmap in the background is computed with Eq. (F.2) for all possible relative positions of three aerial maps. Dark values mean low ZMSSD. The white ellipse shows the covariance of the 200 green particles. The true position is marked with a black cross.

where σ depends on the resolution of the costmap and has been set to 0.08 in our experiments. This results in an initial distribution of the particles that spreads them among the local minima. When a new height-map is available from the MAV, the particles are propagated with the following motion model:

$$\mathbf{u}_t^{[n]} = \mathbf{u}_{t-1}^{[n]} + \Delta\mathbf{u} + \epsilon, \qquad \text{where } \epsilon \sim \mathcal{N}(0, \Sigma). \tag{F.6}$$

The relative motion $\Delta\mathbf{u}$ is provided by the SLAM on the MAV (Section F.4).

Using the cost from Equation (F.2), the propagated particles are weighted and resampled. Hence, the full cost map of the search region is only required for the first aerial map to guarantee a good initial distribution of the particles. In each subsequent step, the cost is only computed at the M particle locations.

In the experiments we found that the particles converge to the true location after maximally three to four iterations (see Figure F.10(a)).

Pose Refinement

Given an initial guess of the relative pose between the MAV and the ground robot, the relative pose can be refined through alignment of the respective pointclouds using ICP [Besl and McKay, 1992]. In order to assure convergence to the global minima, ICP needs to be initialized close to the solution; hence, the global alignment in the section above. Furthermore, the structure of the two pointclouds must be such that their relative movement is constrained (e.g., through both horizontal and vertical structures).

We use the modular ICP algorithm *libpointmatcher* [Pomerleau et al., 2011] that is provided as open-source software. To find the nearest-neighbour points, we apply a kd-tree which is provided by libpointmatcher. As an error metric, we use the point-to-plane distance.

In the experimental-results section, we demonstrate that the pointclouds computed from the dense reconstruction can be aligned with the ground-based map using ICP with an accuracy of 8 cm. Furthermore, we show that the alignment result from the previous section can be refined through ICP.

Experimental Results

We validated the proposed system on four datasets, three were collected indoors and one outdoors. The datasets consist of video and IMU recordings from both, the aerial and ground robot's point of view. The indoor environments were created out of cardboard boxes to resemble a disaster scenario (see Figure F.2). Additionally, the ground-truth robot trajectories were recorded indoors with a motion-capture system. A video of the experimental results is available at http://rpg.ifi.uzh.ch.

SLAM Results

Figure F.9(a) and F.9(b) illustrate exemplary the translation error of the SLAM algorithms on the MAV and the RGBD ground robot respectively. Notice that the trajectory of the MAV does not drift. This is because the MAV flies several loops contrary to the ground robot. The Root-Mean-Square (RMS) error of the Monocular SLAM trajectory is 1.2 cm and for the ground robot 3.8 cm. Average timings of the algorithm are provided in Table F.1. The RGBD SLAM algorithm is slightly slower because on average more map-points were triangulated and tracked.

Dense Reconstruction

The map computed by the monocular SLAM of Section F.4 provides sparse information on the scene observed by the MAV and is conveniently used to determine the extent of the current volume of interest. The set of consecutive views that are aggregated to form a depth map is simply characterised by the distance from the reference camera pose, and controlled by a tuneable threshold parameter set to 12 cm in our experiments. Similarly, a threshold on the distance from which the first depth map has been acquired characterises the set of depth maps to be fused. This distance was set to 70 cm for the experimental validation. Despite the basic view selection strategy controlling depth map creation and fusion, the proposed approach proved highly effective in computing dense and accurate data for the air-ground registration. The λ parameter was set to

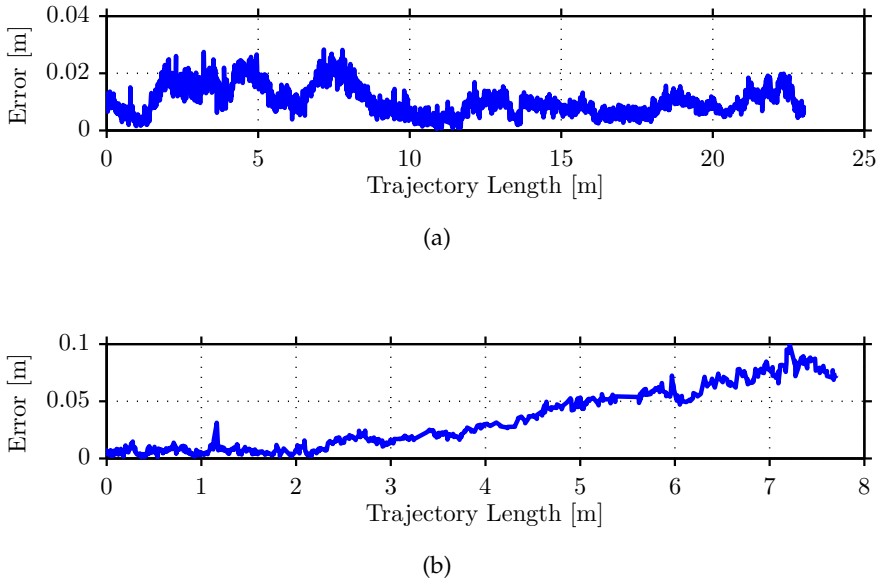
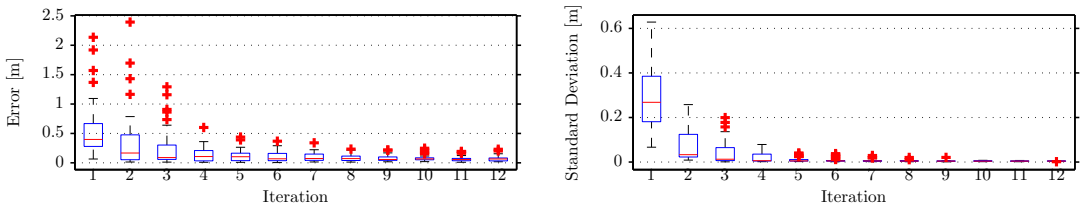


Figure F.9 – (a) Translation error of the Monocular SLAM on the MAV. The trajectory is 23.0 meters long and the RMS error is 1.2 cm. There is no visible drift because the trajectory contains many loops. (b) Translation error of the RGBD SLAM on the ground robot. The trajectory is 7.7 meters and the RMS error is 3.8 cm.

0.26, while 8 iterations proved a good tradeoff between speed and accuracy for the minimisation in Equation F.1.

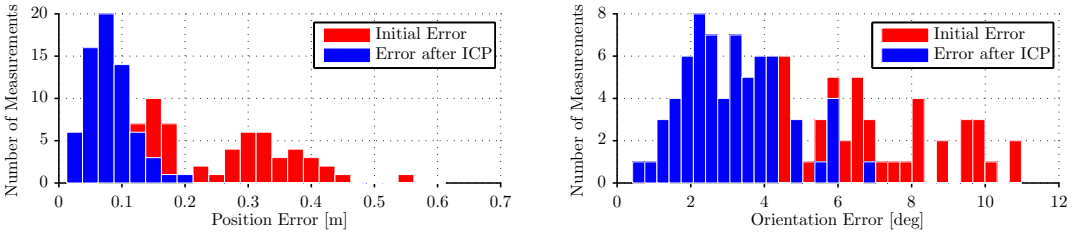
Global Localization

The Monte-Carlo–based localization algorithm was tested on 41 sequences of 12 subsequent depth-maps from three different indoor environments. In Figure F.10(a) the distribution of the localization error is reported for all 12 iterations. In 65% of the experiments, the distance between the mean of the particle distribution at the first iteration and the true position is less than 0.5 meters. Hence, the global minima of the costmap could attract most of the particles. After 4 iterations, the particle means of 89% of the 41 experiments have moved closer than 0.25m to the ground truth. At this range, the ICP algorithm can further refine the pose. Note that the accuracy of the alignment could be further improved by increasing the resolution of the height-maps at the cost of higher computation times. The processing time (Table F.1) for the first frame is approximately 9 seconds for the 4×5 meters search area on a single CPU. Furthermore, for every subsequent iteration, the ZMSSD cost must only be computed at the particle locations. We selected $M = 200$ particles which resulted in a processing time of approximately 38 ms on the CPU. Note that the processing time depends on the size of the depth-map, the search radius, and the number of particles. Furthermore,



(a) (b)

Figure F.10 – (a) Distribution of the translation error over 12 iterations of the Monte-Carlo–based localization illustrated with boxplots. The central mark on the box is the median, the edges of the box are the 25th and 75th percentiles. Results are from 41 experiments. **(b)** Distribution of the standard deviation of the particles (see ellipses in Figure F.8) over 12 iterations of the Monte-Carlo–based localization illustrated with boxplots. Results are from 41 experiments.



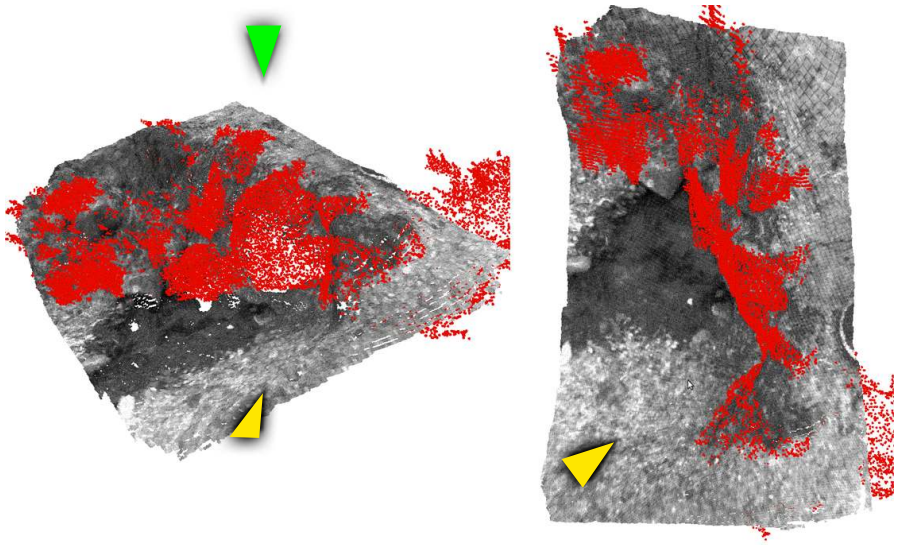
(a) (b)

Figure F.11 – (a) Distribution of the translation error before and after ICP pose refinement. The data originates from 67 experiments in 3 different environments. The median error is 0.076 m. **(b)** Distribution of the angular error before and after ICP pose refinement. The provided angle derives from the angle-axis representation of the orientation error. The data originates from 67 experiments. The median error is 3.0 deg.

it was not necessary to inject new particles after the initial sampling. In order to detect whether the particles have converged, the covariance of the particle distribution can be considered, which is illustrated in Figure F.10(b). One can observe, that with the convergence of the error also the variance decreases.

Pose Refinement

The pose refinement was tested with 67 depth-maps computed from the dense reconstruction in the three indoor environments. The translation and orientation errors before and after the alignement are reported in Figure F.11(a) and F.11(b). Since the monocular SLAM algorithm of the MAV is too accurate to illustrate the performance of the ICP algorithm, we artificially added Gaussian noise with $\sigma_{\text{ang}} = 3deg$ to the orientation, and $\sigma_{\text{trans}} = 0.2m$ to the position. The experiments show that the dense map



(a) (b)

Figure F.12 – Results from the outdoor experiment. Figure (a) shows the aligned maps from the viewpoint of the ground robot (yellow triangle) and Figure (b) the same two maps from the aerial viewpoint (green triangle). The red pointcloud is computed from the ground robot and the dense greyscale pointcloud originates from the reconstruction of the aerial views. Refer to Figure F.2 for two images from the dataset.

computed by the monocular reconstruction is accurate and dense enough to succeed in the alignment with an accuracy of 8 cm and of 3 deg. There are two reasons why the error is not smaller: the ground map by the RGBD SLAM drifts (see Figure F.9(b)) or the error source could come from inaccuracies in the dense reconstruction. The ICP algorithm requires on average 0.5 seconds to align the maps. Note that all depth-maps contained 3D structures, which is a requirement for the ICP algorithm to converge to the global minima, as discussed above. As soon as a map is available from the MAV, pose refinement is run on a dedicated thread. Given an acquisition rate of 30 frames per second, a new augmented map is available approximately every 300 frames.

Remarkably, the complete pipeline is capable of real-time performance on multi-core machines, and the timing for the complete execution is reported in Table F.1.

Outdoor Experiment

Figure F.12 illustrates a result of the outdoor experiment. The same environment is also depicted in Figure F.2. The dense reconstruction algorithm produced qualitatively highly accurate reconstructions due to the naturally very textured surface. The proposed pipeline succeeded in finding the correct alignment of the two maps. The accuracy cannot be reported as no groundtruth is available. Note that in this scenario state-of-the-art algorithms for wide-baseline visual feature maching normally fail as

	Runtime [ms]
Egomotion Estimation:	
Monocular SLAM: Avg. time per frame	14
RGB-D SLAM: Avg. time per frame	15
Dense Reconstruction:	
Add frame to depth map (200 depth hypotheses)	5
Compute distance field from depth map ($376 \times 240 \times 150$ voxels)	21
Depth map fusion (8 iterations)	800
Ray casting	15
Global Localization:	
Full costmap computation (first depth-map)	9143
ZMSSD for $M = 200$ particles:	38
Pose Refinement:	
ICP alignment	462

Table F.1 – Average runtimes of the algorithms in the pipeline. The timings were measured on an 8 core i7 laptop, 2.4 GHz. The used GPU is a Nvidia Quadro K2000M with 384 CUDA cores.

reported in Section F.2.

Conclusion

In this paper, we introduced a method to register the 3D structure computed by a MAV with that computed by a ground robot operating in close range. The MAV is equipped with a monocular camera while the ground robot relies on a range sensor. Building on the recent development of real-time, monocular dense mapping techniques, the proposed method allows the integration of structures computed from radically different viewpoints, i.e. from the aerial and the ground robot. Therefore, this paper contributes a novel approach to the fusion of visual maps with the ones computed from different depth-sensor modalities. Thereby, we prove how dense structure computation from monocular moving cameras is highly valuable in robot perception tasks. We demonstrated the effectiveness of the presented approach in augmenting the three-dimensional structure from the ground robot with an aerial dense map, in two different scenarios: three indoor, experimental setups, and one outdoor, where state-of-the-art alignment methods based on appearance normally fail.

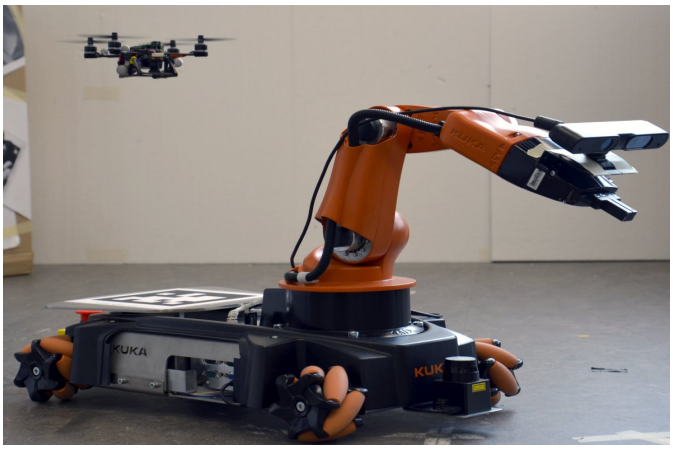


Figure F.13 – The NanoQuad MAV, equipped with a down-looking camera and an onboard computer, is hovering above the Kuka YouBot ground robot, equipped with an RGBD camera.

G Appearance-based Active Dense Reconstruction

Reprinted with permission from:

C. Forster, M. Pizzoli, D. Scaramuzza, “Appearance-based Active, Monocular, Dense Reconstruction for Micro Aerial Vehicles”, Robotics: Science and Systems (RSS), Berkely, 2014.

Appearance-based Active, Monocular, Dense Reconstruction for Micro Aerial Vehicles

Christian Forster, Matia Pizzoli, Davide Scaramuzza

Abstract — In this paper, we investigate the following problem: given the image of a scene, what is the trajectory that a robot-mounted camera should follow to allow optimal dense depth estimation? The solution we propose is based on maximizing the information gain over a set of candidate trajectories. In order to estimate the information that we expect from a camera pose, we introduce a novel formulation of the measurement uncertainty that accounts for the scene appearance (i.e., texture in the reference view), the scene depth and the vehicle pose. We successfully demonstrate our approach in the case of real-time, monocular reconstruction from a micro aerial vehicle and validate the effectiveness of our solution in both synthetic and real experiments. To the best of our knowledge, this is the first work on *active, monocular dense* reconstruction, which chooses motion trajectories that minimize perceptual ambiguities inferred by the texture in the scene.

Introduction

Recent advances in Structure-from-Motion and Visual SLAM made real-time, dense reconstruction from multiple views a viable alternative to laser range finders in robot perception tasks. Impressive results have been demonstrated in the context of Multi-View Stereo (MVS) [Newcombe et al., 2011b, Stühmer et al., 2010, Vogiatzis and Hernández, 2011], where the knowledge of the camera motion is used to estimate depth from different vantage points. Nonetheless, depending on the scene, camera motion

plays a fundamental role in the quality of the obtained reconstruction.

When observing demonstrations of monocular dense reconstruction from hand-held cameras, such as [Newcombe et al., 2011b, Pizzoli et al., 2014], one can notice the commonly used pattern of moving the camera in a *circular trajectory* around a reference view.[1] Intuitively, a circular trajectory constitutes a reasonable approach, as the generated epipolar lines span uniformly the images and increase the chances of reliable stereo matches. Now, suppose that monocular vision is used by a robot to estimate the depth. What radius should we use for the circular camera trajectory? Or more generally, *what is the camera trajectory that provides the best depth measurements?*

In practice, the *best* trajectory depends on different factors: (i) the depth estimate of the scene; (ii) the uncertainty of the current estimate; (iii) the appearance (texture) of the scene; (iv) the current robot pose. Based on the aforementioned considerations, in this paper we introduce a Bayesian formulation to estimate dense depth maps from a Micro Aerial Vehicle (MAV). The next best poses are computed as a function of the robot's current pose and motion as well as the expected depth uncertainty reduction due to predicted future measurements.

A video demonstrating the system is available on the author's website: http://rpg.ifi.uzh.ch.

Related Work

The problem of computing the optimal views to reconstruct an object or a scene has been studied for more than two decades and is known in the computer vision literature as active vision, View Path Planning (VPP), or Next-Best-View (NBV) [Aloimonos et al., 1988, Bajcsy, 1988, Blake and Yuille, 1988, Chen et al., 2011, Scott et al., 2003]. Often, the sensor motion is restricted to a sphere and it is assumed that the object of interest is at all times located completely in the sensor frustum. Proposed algorithms reason about voxel occupancy, occlusion edges, and surface coverage [Huang et al., 2012, Kriegel et al., 2013]. Schmid et al. [Schmid et al., 2012] addressed view planning with an MAV. Similarly to our work, the authors compute a set of aerial views to be used in a multi-view stereo pipeline. However, differently from our approach, their system assumes an *a-priori* model of the scene of interest. Viewpoints are, thus, computed off-line on the pre-computed object hull and the most informative ones are selected on the basis of heuristics that aim at providing full scene coverage. In contrast, we provide an active depth estimation method operating in real-time and on-line.

In the robotics community, a related field to view planning is known as exploration. The first to close the loop between view planning and 3D reconstruction were Whaite

[1] http://youtu.be/Df9WhgibCQA, http://youtu.be/QTKd5UWCG0Q

and Ferrie [Whaite and Ferrie, 1997]. The exploration of a depth-sensor attached to a robot arm was driven by uncertainty reduction of a probabilistic surface model. Feder et al. [Feder et al., 1999] proposed the first work on active SLAM that seeks to minimizes both vehicle and landmark uncertainties. Bourgault et al. [Bourgault et al., 2002] proposed to complement the sparse feature-based SLAM approach with an occupancy grid to provide means of integrating dense range measurements. The proposed exploration policy uses the entropy in the occupancy grid map to stimulate exploration while the uncertainty in the SLAM assures localization accuracy. This approach was extended to particle-filter SLAM [Stachniss et al., 2005] and recently to pose-graph SLAM [Valencia et al., 2012].

While the previous works relied on depth sensors, Davison and Murray [Davison and Murray, 2002] were the first to take into account the effects of actions during *visual* SLAM. The goal was to select a fixation-point of a moving stereo head attached to a mobile robot in order to minimize the motion drift along a predefined trajectory. Vidal Calleja et al. [Vidal-Calleja et al., 2010] demonstrated an active feature-based visual SLAM framework that provides real-time user-feedback to minimize both map and camera pose uncertainty. Bryson and Sukkarieh [Bryson and Sukkarieh, 2008] demonstrated a similar visual and inertial EKF-SLAM formulation for active control of flying vehicles. The goal was to cover a predefined area with a camera sensor while maintaining an accurate estimation of both the map and the vehicle state. Extensive simulation results were provided of a MAV that is restricted to fly on a plane. Similar to [Vidal-Calleja et al., 2010, Bryson and Sukkarieh, 2008] the exploration in our algorithm is driven by a set of states (i.e., dense depth estimates in the reference view) that are initialized with high uncertainty at the start of the exploration. Our resulting map is spatially smaller but denser and exhibits more detail, which is crucial e.g., for path planning in cluttered environments. Furthermore, in [Vidal-Calleja et al., 2010, Bryson and Sukkarieh, 2008] the image is only used to extract features and subsequently neglected. On the other hand, our proposed approach is *direct* [Irani and Anandan, 1999]—the intensity values in the image are directly used to reason about the next best view.

In [Soatto, 2009], Soatto introduces the notion of *Actionable Information* that is the portion of data that is useful towards the accomplishment of a task and after discounting nuisance factors. In [Soatto, 2011, Chapter 8], he describes a hypothetic greedy explorer that tries at every time instant to maximize the *Actionable Information Increment* (AIN). He argues that such an explorer can get stuck in a local minima where no control action yields any information and, therefore, suggests two improvements: firstly, to plan a trajectory that maximizes the AIN over a *finite horizon*. Secondly, to use the *memory* of past observations to build a representation of the environment and to plan the trajectory so as to minimize the uncertainty in this representation. Soatto recognizes that it is trivial to design an explorer that achieves complete exploration of a static environment as, for instance, a random explorer (Brownian motion) would asymptotically do so.

However, the goal is to do so *efficiently*. In this work we present an implementation of such an explorer for monocular, dense depth estimation.

Contributions and Outline

State-of-the-art approaches to active mapping [Kriegel et al., 2013, Bourgault et al., 2002, Davison and Murray, 2002, Stachniss et al., 2005, Valencia et al., 2012, Sim and Roy, 2005] retain only geometric information while discarding the scene appearance. As a result, a robot trying to perceive the depth of a white wall, would generate different camera trajectories in vain, eventually failing to reduce the uncertainty in the depth measurement [Soatto, 2009]. By contrast, we propose a method to compute the measurement uncertainty and, thus, the expected information gain, on the basis of scene structure *and* appearance (i.e., texture). By doing so, surfaces characterized by uniform intensity yield high uncertainty in stereo computation, thus encoding the fact that there is no information to obtain from staring at white walls.

The contributions introduced by this paper can be summarized as follows.

- We propose a formulation of the uncertainty characterizing a depth measurement from multi-view stereo that takes into account the appearance of the scene, the motion of the camera, and the structure of the scene currently available. This formulation is used to evaluate candidate camera poses on the basis of the expected information gain.

- For applications to dense reconstruction from MAVs, we provide a strategy to compute a candidate sequence of viewpoints that lie on a feasible trajectory and that maximize the expected information gain.

- We detail both synthetic and experimental validation of the proposed system in closed loop and compare against four different control strategies: a random strategy, a circular motion, a greedy strategy and a Next-Best-View (NBV) strategy that iteratively selects the globally optimal view points.

The outline of the paper follows. In Section G.2 we detail our method to compute probabilistic depth maps from a moving camera, introduce our evaluation method and optimality criterion. Section G.3 presents different strategies for the generation of candidate trajectories and Section G.4 is dedicated to the discussion on the experimental evaluation. Finally, in Section G.5, we summarize our contribution and draw the conclusions.

Probabilistic Monocular Depth Estimation

In this section, we formalize the recursive Bayesian estimation of depth from multi-view stereo, focusing on the measurement uncertainty, which is the crucial factor for planning informative trajectories.

We denote the intensity image collected at time step k as $\mathtt{I}_k : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$, where Ω is the image domain. Let the rigid-body transformation $\mathtt{T}_{w,k} \in SE(3)$ describe the pose of the camera acquiring $\mathtt{I}_k$ in the *world* reference frame. The inverse depth $\hat{d}_{\mathbf{u}}$ of a pixel $\mathbf{u}$ in the *reference* camera pose $\mathtt{T}_{w,r}$ is a latent variable we infer from observations. An observation is a pair $\{\mathtt{I}_k, \mathtt{T}_{w,k}\}$, where we assume that $\mathtt{T}_{w,k}$ is computed by an accurate visual odometry algorithm [Forster et al., 2014b]. A measurement $d_{\mathbf{u},k}$ of pixel $\mathbf{u}$ is obtained by the k-th observation by triangulating from $\mathtt{T}_{r,k} = \mathtt{T}_{w,r}^{-1} \cdot \mathtt{T}_{w,k}$ and we assume it normally distributed with mean $\mu_{\mathbf{u},k}$ and variance $\tau^2_{\mathbf{u},k}$:

$$p(d_{\mathbf{u},k}|\hat{d}_{\mathbf{u}}) = \mathcal{N}(d_{\mathbf{u},k}|\mu_{\mathbf{u},k}, \tau^2_{\mathbf{u},k}). \tag{G.1}$$

Given a prior $p(\hat{d}_{\mathbf{u}})$ and assuming independent and identically distributed measurements, the estimation proceeds recursively from the observations $k \in \{r+1, \dots, n\}$:

$$p(\hat{d}_{\mathbf{u}}|d_{\mathbf{u},r+1}, \dots, d_{\mathbf{u},n}) \propto p(\hat{d}_{\mathbf{u}}) \prod_{k=r+1}^{n} p(d_{\mathbf{u},k}|\hat{d}_{\mathbf{u}}). \tag{G.2}$$

Upon the k-th observation, the posterior (G.2) is normally distributed with parameters computed from the estimation at time $k-1$:

$$\begin{aligned} &p(\hat{d}_{\mathbf{u}}|d_{\mathbf{u},r+1}, \dots, d_{\mathbf{u},k}) = \mathcal{N}(\hat{d}_{\mathbf{u}}|\mu_{\mathbf{u},k}, \sigma^2_{\mathbf{u},k}), \text{ with} \\ &\mu_{\mathbf{u},k} = \frac{\sigma^2_{\mathbf{u},k-1} d_{\mathbf{u},k} + \tau^2_{\mathbf{u},k} \mu_{\mathbf{u},k-1}}{\sigma^2_{\mathbf{u},k-1} + \tau^2_{\mathbf{u},k}}, \quad \sigma^2_{\mathbf{u},k} = \frac{\sigma^2_{\mathbf{u},k-1} \tau^2_{\mathbf{u},k}}{\sigma^2_{\mathbf{u},k-1} + \tau^2_{\mathbf{u},k}}. \end{aligned} \tag{G.3}$$

A similar model to estimate the depth of a pixel is used in [Pizzoli et al., 2014, Vogiatzis and Hernández, 2011]. To increase the robustness of this approach, it is proposed in [Vogiatzis and Hernández, 2011] to explicitly model outliers. Furthermore, in [Pizzoli et al., 2014], we showed how regularity in the depth map can be enforced by making use of a smoothness prior in regions characterized by high uncertainty.

Measurement uncertainty

A camera is a passive sensor and the measurement uncertainty is a function of the depth, the camera motion, and the scene texture. In this section, we detail how to compute the measurement uncertainty τ_k related to a candidate camera motion $\mathbf{T}_{r,k}$, starting

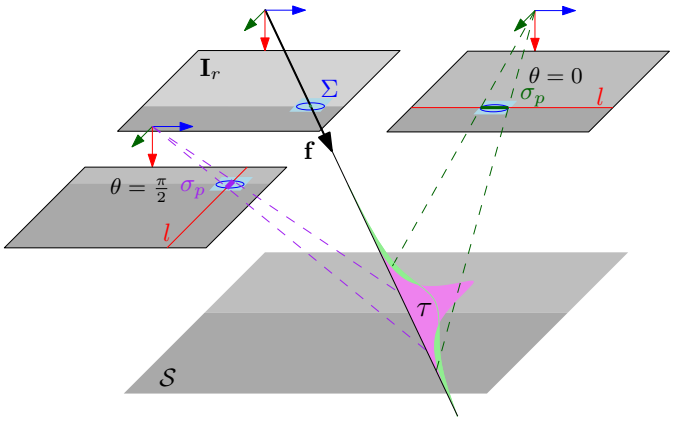


Figure G.1 – Disparity uncertainty. Depending on the image gradient, the camera motion influences the reliability of stereo matching and, thus, the uncertainty in the disparity computation σ_p^2.

from estimating the photometric disparity uncertainty $\sigma_{p,k}$, which accounts for possible ambiguities in epipolar matching (e.g., due to uniform texture), and propagating it through triangulation to the depth uncertainty τ_k.

The disparity error accounts for uncertainty in disparity measurement given the reference image appearance $\mathbf{I}_r$ and the camera motion $\mathbf{T}_{r,k}$. It encapsulates the fact that some motions are better than others to compute the disparity related to a pixel. Indeed, the camera motion determines the direction of the epipolar line l and the disparity measurement relies on comparison of intensity patches. Intuitively, matching is reliable for image patches characterized by strong intensity gradients; in the context of active vision, this means that the direction of the gradient in a region must be considered in order to select a motion that is suitable for the disparity estimation. For instance, when reconstructing regions characterized by a dominant gradient direction (see Figure G.1), a camera motion resulting into epipolar lines that are parallel to the dominant gradient direction in the intensity image (e.g., motion to the right in Figure G.1) will result in less reliable epipolar matches and, thus, higher uncertainty in the disparity σ_p and subsequently in depth τ.

More precisely, when the *sum of squared differences* (SSD) between image patches is used for stereo matching, the probability of a correct match in the neighbourhood of a pixel can be expressed by a zero mean bivariate normal distribution [Matthies et al., 1988], with covariance matrix

$$\mathbf{\Sigma} = 2\sigma_i^2(\mathbf{J}\mathbf{J}^\top)^{-1}, \tag{G.4}$$

where we denote by σ_i^2 the variance of the image noise and by $\mathbf{J} = \sum_P (\partial\mathbf{I}/\partial x, \partial\mathbf{I}/\partial y)$ the sum of the image gradients over a patch P, centered at the pixel of interest.

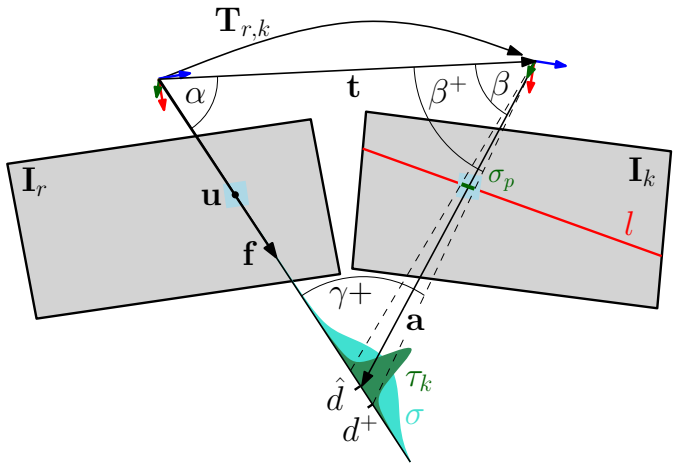


Figure G.2 – The uncertainty in depth measurement, τ_k^2, is computed by projecting the disparity uncertainty σ_p in image $\mathbf{I}_k$ on the pixel bearing-vector $\mathbf{f}$.

We now take into account the camera motion and derive the uncertainty of disparity computation when matching is performed along the epipolar line generating from $\mathbf{T}_{r,k}$. Let θ be the angle formed by the epipolar line and the image x axis. We can transform the probability of a correct match to a reference system that has the x axis aligned with the epipolar line, which results in a covariance matrix

$$\boldsymbol{\Sigma}' = \left(\mathbf{R}^\top \boldsymbol{\Sigma}^{-1} \mathbf{R}\right)^{-1}, \quad \mathbf{R} = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}. \tag{G.5}$$

The disparity error along the epipolar line follows the conditional distribution $p(x|y = 0)$, which is Gaussian and characterized by the variance (cfr. [Bishop, 2006, p.87])

$$\sigma_p^2 = \Sigma'_{xx} - \Sigma'_{xy}\Sigma'^{-1}_{yy}\Sigma'_{yx}, \tag{G.6}$$

where Σ'_{xx}, Σ'_{xy} and Σ'_{yy} are the entries of $\boldsymbol{\Sigma}'$.

Thus, the disparity error is normally distributed along the epipolar line with variance

$$\sigma_p^2 = \frac{|\boldsymbol{\Sigma}|}{\Sigma_{xx}\sin^2(\theta) + 2\Sigma_{xy}\sin(\theta)\cos(\theta) + \Sigma_{yy}\cos^2(\theta)}, \tag{G.7}$$

where Σ_{xx}, Σ_{xy} and Σ_{yy} correspond to entries of $\boldsymbol{\Sigma}$ and $|\boldsymbol{\Sigma}|$ is the determinant of $\boldsymbol{\Sigma}$.

In the active vision context, we cannot compute the disparity error on the new image, as the image is not available at the time we predict the measurement uncertainty. Therefore, we consider the epipolar line in the reference image and compute the

disparity error therein. The assumption that the patch appearance can be predicted by the reference patch is valid for small viewpoint changes (i.e., neglecting distortions and occlusions).

The measurement variance of the depth at pixel $\mathbf{u}$ in the image $\mathbf{I}_k$ is obtained by back-projecting the variance of the photometric disparity error σ_p^2. Referring to Figure G.2, let $\hat{d}$ be the current depth estimation of pixel $\mathbf{u}$, the corresponding unit bearing vector is denoted as $\mathbf{f}$ and $\mathbf{t}$ denotes the translation component of the relative position $\mathrm{T}_{r,k}$. As proposed in [Pizzoli et al., 2014], we can transform the measurement uncertainty σ_p^2 in the image to the depth uncertainty τ_k^2 as follows:

$$\mathbf{a} = \hat{d} \cdot \mathbf{f} - \mathbf{t}, \qquad \alpha = \arccos\left(\frac{\mathbf{f} \cdot \mathbf{t}}{||\mathbf{t}||}\right), \qquad \beta = \arccos\left(-\frac{\mathbf{a} \cdot \mathbf{t}}{||\mathbf{a}|| \cdot ||\mathbf{t}||}\right). \tag{G.8}$$

Let f be the camera focal length. The angle spanning σ_p pixels can be added to β in order to compute γ^+ and, thus, by applying the law of sines, recover d^+:

$$\beta^+ = \beta + 2\tan^{-1}\left(\frac{\sigma_p}{2f}\right), \qquad \gamma^+ = \pi - \alpha - \beta^+, \qquad d^+ = ||\mathbf{t}||\frac{\sin\beta^+}{\sin\gamma^+}. \tag{G.9}$$

Therefore, the measurement uncertainty is computed as:

$$\tau_k^2 = \left(d^+ - \hat{d}\right)^2. \tag{G.10}$$

The derivation of the depth uncertainty reported in Equations (G.8) - (G.10) is similar to the one presented in [Pizzoli et al., 2014], however with one critical difference that occurs in Equation (G.9). In the present paper, the disparity uncertainty σ_p is a function of the *appearance* (i.e., texture) in the scene. In contrast, in [Pizzoli et al., 2014] this was simply set to 1, meaning that the uncertainty was assumed independent of the scene appearance.

The Information Gain of a Measurement

We now demonstrate how the proposed probabilistic depth map representation and update method can be applied to the problem of selecting the next best placements for the camera.

Suppose that we are computing the depth map for a given reference image $\mathbf{I}_r$. We describe the uncertainty in the depth map estimate at time k with the entropy $\mathcal{H}_k$. In such a way, the treatment is independent on the actual model and the parametric formulation described in Section G.2 might be replaced in order to take into account, for instance, multiple depth hypotheses [Wendel et al., 2012].

Since, for every pixel $\mathbf{u} \in \Omega$, the depth estimation proceeds independently, $\mathcal{H}_k$ can be computed as (see, for instance, [Bishop, 2006])

$$\mathcal{H}_k = \frac{1}{2} \sum_{\mathbf{u} \in \Omega} \ln(2\pi e \, \sigma^2_{\mathbf{u},k}), \tag{G.11}$$

where $\sigma^2_{\mathbf{u},k}$) denotes the depth uncertainty of pixel $\mathbf{u}$ at time k (see Eq. (G.3)).

Upon the acquisition of a measurement from the $(k+1)$-th camera pose $\mathbf{T}_{r,k+1}$, the variance of the estimated depth for the pixel $\mathbf{u}$ is updated to take into account the measurement uncertainty $\tau^2_{\mathbf{u},k+1}$.

We define the *information gain* as the difference

$$\mathcal{I}_{k,k+1} = \mathcal{H}_k - \mathcal{H}_{k+1}, \tag{G.12}$$

which, plugging (G.3) into (G.11), yields

$$\mathcal{I}_{k,k+1} = \frac{1}{2} \sum_{\mathbf{u} \in \Omega} \ln \left\{ \frac{\tau^2_{\mathbf{u},k+1} + \sigma^2_{\mathbf{u},k}}{\tau^2_{\mathbf{u},k+1}} \right\}. \tag{G.13}$$

Solution Strategies

In this section, we describe five different control strategies for the active depth-map estimation problem. The control strategies range from random, heuristic, and greedy methods to a model-predictive control approach that optimizes the next N views to maximize the information gain. In Section G.4, we will evaluate the proposed methods in synthetic and real-world experiments.

We simplify the problem by assuming that the camera moves at constant speed and takes measurements at fixed frame rate. This results in equidistant measurements with a relative distance $\Delta\mathbf{t} \in \mathbb{R}^3$ that is fixed a priori. The proposed system can be extended to incorporate the inertia, controllability, and the dynamics of the camera-equipped robot.

One can obtain more precise, thus more informative, measurements closer to the surface. Therefore, an optimal control strategy eventually would make the robot approach the surface (see Figure G.3 (b)). To avoid collisions in our envisioned MAV application, we additionally restrict the motion to the horizontal plane $\mathcal{Z}$ at the height of the reference view. Nevertheless, all proposed solution strategies can be extended to the 3D space with increased computational cost that comes with the enlarged action space.

With these assumptions we can formalize the problem as follows: given the current pose relative to the reference view $\mathtt{T}_{r,k}$ and the proposed method to measure the

information gain of a measurement at the next pose $\mathcal{I}_{k,k+1} = \mathcal{I}_{k,k+1}(\mathtt{T}_{k,k+1})$, which next pose $\mathtt{T}_{r,k+1} \in \mathcal{A}_k$ should be selected? The action space at time k is defined such that equidistant camera poses in the horizontal plane are selected:

$$\mathcal{A}_k = \left\{ \mathtt{T} \;\middle|\; ||\mathtt{T}_{r,k}^{-1} \cdot \mathtt{T}||_2 = \Delta\mathbf{t} \;\wedge\; \mathtt{T} \in \mathcal{Z} \right\}. \tag{G.14}$$

Random Walk Control

Similar to [Soatto, 2009], we use as a baseline a random walk strategy that at every measurement k selects randomly the next pose from the action space $\mathcal{A}_k$. This approach is completely blind, hence should perform worse than all of the following strategies.

Circular Heuristic Control

A circular trajectory guarantees that the epipolar line sweeps over all directions. Thereby, depth uncertainties that arise from the aperture problem during triangulation can be disambiguated. For this reason, a circular trajectory is intuitively a good heuristic and typically used in demonstrations of monocular multi-view stereo systems [Newcombe et al., 2011b]. However, the radius of the circle must be tuned to the depth of the scene. The radius should trade off accuracy through increased base-line versus visibility of the reconstructed surface area $\mathcal{S}$. In the synthetic experiments we selected the radius to give the best results in the first scene and kept the radius fixed for the other experiments.

Greedy Control

A greedy controller tries to take control actions so as to maximize the expected information gain of the next measurement [Feder et al., 1999]. The greedy control can then be written as follows:

$$\mathtt{T}_{r,k+1} = \arg\max_{\mathtt{T} \in \mathcal{A}_k} \mathcal{I}^*_{k,k+1}(\mathtt{T}). \tag{G.15}$$

This control law is equal to a gradient descent algorithm with fixed step size. Unless the underlying functional is convex or the cost is extended with an additional *curiosity*-term that promotes exploration of unknown areas [Bourgault et al., 2002], this approach is prone to get stuck in local minima.

Next-Best-View Control

Since the information gain proposed in Section G.2.2 can be evaluated not only in the neighbourhood of the current pose but also for all feasible positions and orientations,

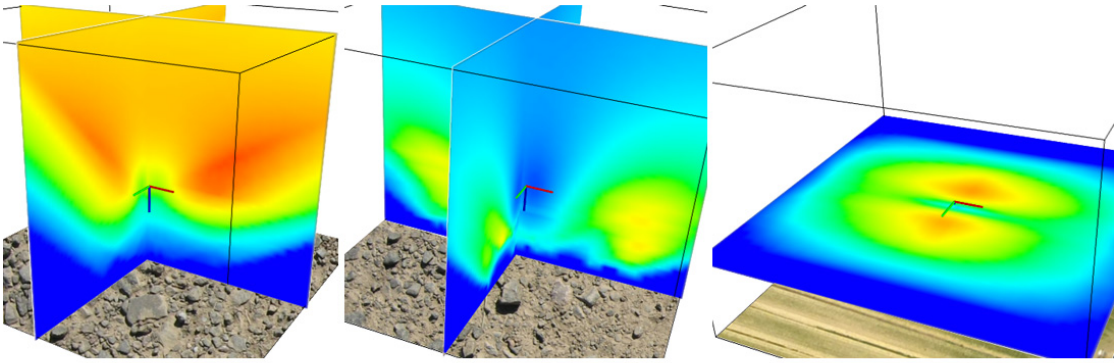


Figure G.3 – Information gain for the NBV strategy. The distributions are visualized as heat-maps (red means high information gain, blue low). Figure (a) shows the information gain before the first measurement in an environment of isotropic gravel texture. Figure (b) shows the information gain after the 10th measurement in the same scene. Figure (c) shows the information gain in an environment with a dominant gradient direction in the texture.

the NBV control always selects the viewpoint in the horizontal plane $\mathcal{Z}$ that provides the highest information gain, independently of the current pose:

$$\mathtt{T}_{r,k+1} = \arg\max_{\mathtt{T}\in\mathcal{Z}} \mathcal{I}^*_{k,k+1}(\mathtt{T}). \tag{G.16}$$

Thus, the translation between poses is not limited to $\Delta\mathbf{t}$ anymore. Since there is no guarantee that subsequent measurements are spatially close, the travel distance of this approach between two measurements will be high.

Receding-Horizon Control

Let us assume that the position of the next N poses $\{\mathtt{T}_{r,k+1}, \ldots, \mathtt{T}_{r,k+N}\}$ can be parametrized by the parameter vector $\boldsymbol{\phi}_k$ such that each pose lies in the action space of the previous pose: $\mathtt{T}_{r,k+i} \in \mathcal{A}_{k+i-1}$. We can improve the greedy control strategy by considering the information gain over the finite horizon N as proposed in [Huang et al., 2005, Soatto, 2011]. Given the current frame k, the receding-horizon control maximizes the expected information gain over the course of the next N views:

$$\boldsymbol{\phi}_k = \arg\max_{\boldsymbol{\phi}} \sum_{i=k}^{k+N} \mathcal{I}^*_{i,i+1}(\boldsymbol{\phi}). \tag{G.17}$$

One can predict the probability of a measurement at time $k+1$ based on the uncertainty in the current depth-map. To compute the expected measurement at time $k+2$ would require to integrate over all possible depth-maps that can result from the update at $k+1$. This problem can be formulated with a partially observable Markov decision

process (POMDP [Pack Kaelbling et al., 1998]) which becomes intractable with high state- and action-spaces.

However, as proposed in [Huang et al., 2005], we can make the assumption that the next measurements do not provide any new *evidence*, meaning that the prediction coincides with the measurement and thus, the mean of the estimate does not change. With this assumption it is straightforward to compute the information gain over the next N measurements:

$$\begin{aligned} &\mathcal{I}^*_{k,k+N} = \mathcal{H}_k - \mathcal{H}^*_{k+N}({\sigma^*}^2_{k+N}), \qquad \text{with} \\ &\frac{1}{{\sigma^*}^2_{k+N}} = \frac{1}{\sigma^2_k} + \frac{1}{{\tau^*_{k+1}}^2(\boldsymbol{\phi}_k)} + \ldots + \frac{1}{{\tau^*_{k+N}}^2(\boldsymbol{\phi}_k)}, \end{aligned} \tag{G.18}$$

where ${\tau^*_{k+i}}^2(\boldsymbol{\phi}_k)$ is the predicted measurement uncertainty at pose $\mathrm{T}_{r,k+i}$ that is a function of the trajectory parameters $\boldsymbol{\phi}_k$.

Increasing the prediction horizon N in this formulation makes sense only when the depth uncertainty is not too high, since this approach is based on the assumption that the mean of the current depth estimate does not change over the next N measurements. Further, note that similarly to the greedy approach, there is no guarantee that this approach does not fall into a local minima.

A heuristic that we apply in order to increase the prediction accuracy in uncertain depth maps and to avoid local minima is to start with a short prediction horizon $N = 3$ when the map is uncertain and to increase the prediction horizon when the predicted information gain $\mathcal{I}^*_{k,k+N}$ falls below some threshold in order to escape local minima. Furthermore, since the depth estimate changes as soon as the $(k+1)$-th measurement is acquired, the trajectory until measurement $N+1$ is replanned immediately.

The computational demand of the prediction grows exponentially with the degrees of freedom of the trajectory parameters $\boldsymbol{\phi}$ and linearly with the prediction horizon N.

Implementation Details

In this section, we provide more details on our implementation of the receding-horizon control strategy and the information gain computation.

To favor the dynamics of the MAV, we reduce the dimensionality of the action space by enforcing the continuity of the trajectory and by setting the tangent at the current position to the current direction of motion. Additionally, we prohibit yaw camera rotations in order to minimize motion blur. We chose to parametrize the trajectory with a B-spline [de Boor, 2001] of third-order with three control points (see Figure G.4). B-splines are piecewise polynomial functions with local support and simple derivatives.

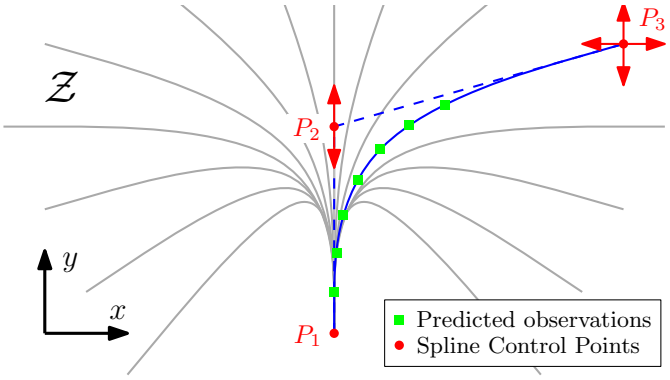


Figure G.4 – B-Spline trajectory parametrization. P_1, P_2 and P_3 are the control points. The candidate camera poses are visualized in green.

However, any other temporal basis function could be used. The first control point P_1 of the B-spline is set fixed to the current position of the camera, the second control point P_2 has one degree of freedom (P_{2y}) along the current direction of motion of the MAV and the third control point P_3 has two degrees of freedom in the horizontal plane $\mathcal{Z}$ (P_{3x} and P_{3y}, see Figure G.4). In total the trajectory parametrization has three degrees of freedom $\boldsymbol{\phi} = \{P_{2y}, P_{3x}, P_{3y}\}$. By setting constraints on the position of $\{P_1, P_2\}$, it is possible to enforce the dynamic constraints of the MAV on the trajectory. The predicted observations are located along the trajectory with equal distance $\Delta\mathbf{t}$. The optimal trajectory in the three dimensional space can be found by a global optimization routine with the condition that the spline parameters $\boldsymbol{\phi}$ must remain in the range $\pm 2N\Delta\mathbf{t}$.

The computation of the depth-map entropy, which is evaluated multiple times in every control iteration according to (G.11), requires summation over all pixels in the image. To maintain real-time performance, we were required to select a subset of pixels for which the information gain is computed. In practice we compute the information gain, thus, the trajectory, based on 400 uniformly distributed pixels with high gradient magnitude.

Experimental Evaluation

Simulation Experiments

We evaluated the proposed control strategies in three different synthetic environments (Figure G.6(a) to G.7). The scenes vary in both the texture and shape of the surface. Scenes 1 and 2 contain isotropic gravel texture while the texture of Scene 3 exhibits a dominant gradient direction. The surface in Scene 1 and 3 is planar and in Scene 3 there is a step.

To give an intuition of the information distribution, we sampled the information gain regularly in a cube around the reference view and display the results in Figure G.3. The information density before the first measurement in Scene 1 is displayed in Figure G.3 (a) and after the 10th measurement in Figure G.3 (b). The coordinate frame displayed in the center of the figures illustrates the position of the downward-looking reference view. Hot (red) colors indicate relative positions with high expected information gain and cold (blue) colors positions with low potential. Neglecting the restriction of the motion to a horizontal plane for now, one can observe that for the first measurement a horizontal and vertical motion would be optimal. Moving horizontally increases the baseline and moving vertically ensures that the whole surface remains within the field of view. This illustrates intuitvely why planning multiple steps ahead is superior to next-best-view planning: rather than moving upwards and ensuring that the whole depth-map is within the field of view, two close measurements—each updating one side of the depth-map—would result in higher uncertainty reduction. Figure G.3 (b) shows that after a 10 measurements, the information-gain is generally lower and that it is advantageous to move closer to the surface. Figure G.3 (c) shows the initial cost in the horizontal plane of Scene 3. Scenes with isotropic texture exhibit a circular region around the reference view with high information gain. However, since Scene 3 is textured with a dominant gradient direction, the photometric disparity error is higher for motions along the gradient direction (aperture problem). This reflects in the information gain computation and thus motions rectangular to the gradient directions are favoured.

Figure G.8 shows the information gain in the horizontal plane centered two meters above a horizontally striped surface. When neglecting the texture (i.e., $\sigma_p^2 = 1$), the robot would prefer a horizontal motion since less pixels move out of the field of view. However, when considering the appearance, a motion in x direction does not provide any information due to the aperture problem.

The plots in Figures G.6(a) to G.7 compare the proposed control strategies for each of the synthetic environments. The simulation of all control strategies was run until an accuracy of less than 1 mm in the depth-map was reached. The red plane in each rendering illustrates the altitude to which the camera was resticted to move. The reference view is acquired in the center of each red plane with a downward-looking camera. Plot (b) in each figure shows the resulting trajectories on the horizontal plane for all control strategies while Plot (c) shows the entropy reduction over travelled distance. When comparing the information gain over the travelled distance in Plot (c), the greedy approach performs similar to the spline-based method in terms of entropy reduction over travelled distance in the first environment. However, in the second and third environment, the greedy approach gets stuck in a local minimum. The spline-based receding-horizon control requires in all environments the least motion to achieve the predefined accuracy level. The results of the random strategy are averaged over 100 measurements of which we display only one in the trajectory plots.

In Figure G.7 (b) it is clearly visible how the photometric disparity uncertainty drives the receding-horizon control to select views which do not suffer from the aperture problem. After moving in positive y direction, the MAV seems to get stuck in a local minimum, however, by increasing the prediction horizon it finds the path towards the other side of the map.

Real-World Experiments

In Figure G.5(a), we show the setup of the real experiments. The MAV is equipped with a downward-looking camera and embedded processor. A vision-based SLAM algorithm [Forster et al., 2014b] runs onboard to estimate the egomotion and stabilize the vehicle. To achieve real-time performance, we run the dense reconstruction and path planning off-board on an Intel i7 laptop. Therefore, the MAV streams video and estimated poses to a ground-station where the proposed algorithms compute and return in real time the trajectory commands. A video of the experiment can be viewed on the author's website: http://rpg.ifi.uzh.ch.

We compared the three best performing control strategies and report the results Figure G.5(e). In Figure G.5(d), the resulting trajectories are shown, where we additionally display the B-splines that are computed at every iteration. The final depth-map of the spline strategy is shown in Figures G.5(b) and G.5(c).

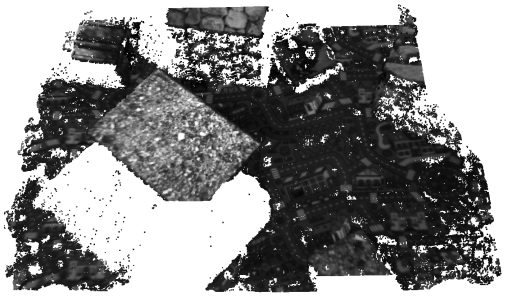
A comparison of the control strategies in real experiments is more challenging than in simulation since the reference view must be taken exactly at the same location, which is almost impossible. For this reason, the comparison of the convergence speed must be analyzed with caution. The greedy method fell in a local minimum and approached a wall when the experiment had to be stopped. For the circle strategy we tuned the radius to give best performance in this scenario. Indeed, it converges slightly faster than the receding-horizon (spline) strategy. The advantage of the spline strategy, however, is that it must not be adapted to the environment height, shape and appearance.

Conclusion and Future Work

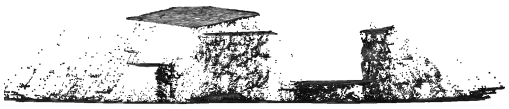
In this paper, we proposed an approach to actively acquire informative views for monocular dense depth estimation. In evaluating a candidate camera trajectory, we proposed to take into account the texture of the scene, and we contributed a novel formulation of the depth measurement uncertainty based on propagating the uncertainty in photometric stereo disparity to triangulation. We evaluated different strategies in both simulation and real scenarios and we showed how the camera trajectories emerging from the information maximization problem are, at the same time, *informative*, in terms of depth estimation, and *parsimonious*, in terms of traveled distance. For applications to

Micro Aerial Vehicle (MAV) perception, we reduced the dimensionality of the search space by enforcing continuity on the trajectory. To the best of our knowledge, this is the first work on active, monocular *dense* reconstruction demonstrated on a robot.

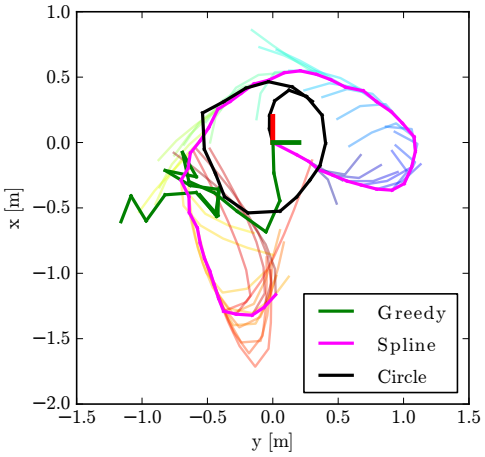
(a) Experiment Setup



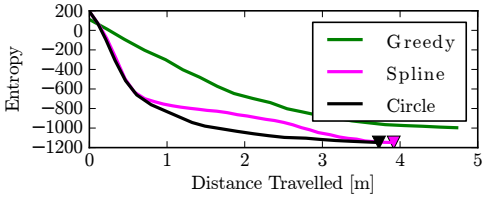
(b) Reconstruction result



(c) Reconstruction result

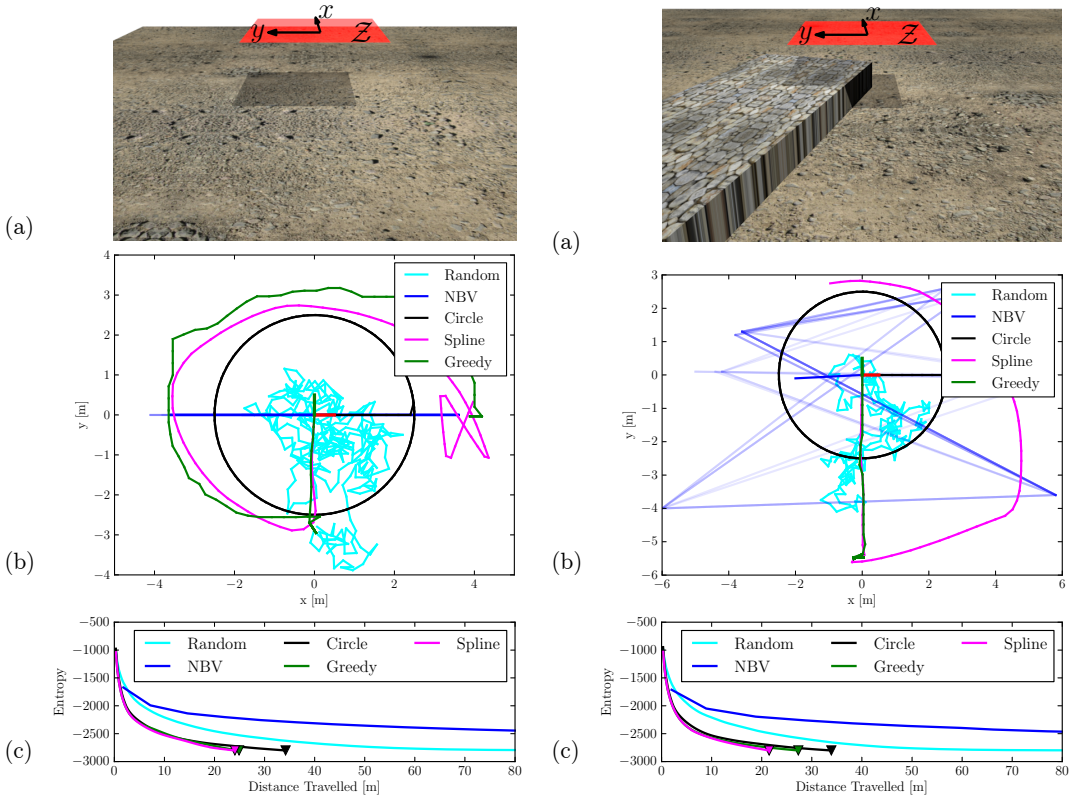


(d) Reconstruction trajectory



(e) Entropy reduction over travelled distance

Figure G.5 – Real world experiment and reconstruction results.



(a) Synthetic scene 1. (b) Synthetic scene 2.

Figure G.6

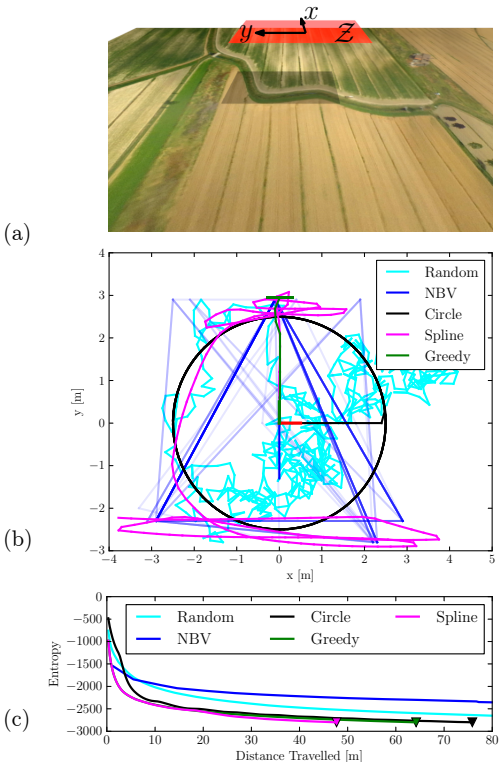
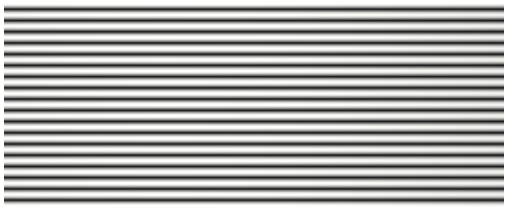
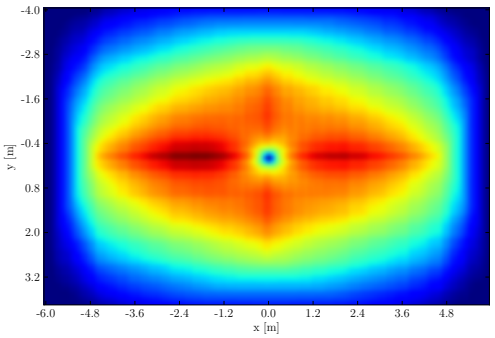


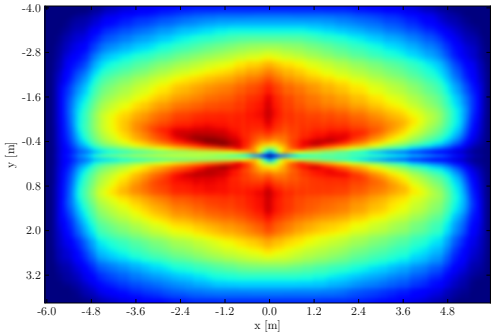
Figure G.7 – Synthetic scene 3.



(a) Horizontally striped texture.



(b) Information gain neglecting the texture.



(c) Information gain using the texture.

Figure G.8 – Influence of striped texture on the information gain.

Bibliography

Adis IMU specifications. URL http://www.analog.com/media/en/technical-documentation/data-sheets/ADIS16448.pdf. 119

Tango IMU specifications. URL http://ae-bst.resource.bosch.com/media/products/dokumente/bmx055/BST-BMX055-FL000-00_2013-05-07_onl.pdf. 119

P. A. Absil, C.G. Baker, and K.A. Gallivan. Trust-region methods on Riemannian manifolds. *Foundations of Computational Mathematics*, 7(3):303–330, 2007. 89, 93, 94

M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy. Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments. In *SPIE Conf. on Unmanned Systems Technology*, 2009. 6

M. Achtelik, S. Weiss, M. Chli, F. Dellaert, and R. Siegwart. Collaborative stereo. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011. 29

A. Agarwal, K. Mierle, and Others. Ceres solver. http://ceres-solver.org. 8, 47

S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. Seitz, and R. Szeliski. Building Rome in a day. *Commun. ACM*, 54(10):105–112, October 2011. ISSN 0001-0782. doi: 10.1145/2001269.2001293. URL http://doi.acm.org/10.1145/2001269.2001293. 3

J. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active vision. *Int. J. Comput. Vis.*, 1(4):333–356, 1988. 189

M. S. Andrle and J. L. Crassidis. Geometric integration of quaternions. *Journal of Guidance, Control, and Dynamics*, 36(6):1762–1757, 2013. 99

R. Bajcsy. Active perception. *Proc. of the IEEE*, 76(8):966–1005, 1988. 189

S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *Int. J. Comput. Vis.*, 56(3):221–255, 2004. 55, 80

Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications To Tracking and Navigation*. John Wiley and Sons, 2001. 8, 47, 50, 76, 109, 111

T. D. Barfoot. *State Estimation for Robotics - A Matrix Lie Group Approach*. Cambridge University Press, 2015. 78, 80

T. D. Barfoot and P. T. Furgale. Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Trans. Robotics*, 30(3):679–693, 2014. 92

B. M. Bell and F. W. Cathey. The iterated kalman filter update as a gauss-newton method. *IEEE Transactions on Automatic Control*, 38(2):294–297, 1993. 87

S. Benhimane and E. Malis. Integration of euclidean constraints in template based visual tracking of piecewise-planar scenes. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2006. 49

P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(2):239–256, 1992. 179

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., 2006. 194, 196

A. Blake and A. Yuille. *Active Vision*. the MIT Press, 1988. 189

M. Bloesch, S. Weiss, D. Scaramuzza, and R. Siegwart. Vision based MAV navigation in unknown and unstructured environments. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2010. 29

M. Bloesch, S. Omari, M. Hutter, and R. Siegwart. Robust visual inertial odometry using a direct EKF-based approach. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, 2015. 87

S. Bosch, S. Lacroix, and F. Caballero. Autonomous detection of safe landing areas for an uav from monocular images. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2006. 151, 152, 153

F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte. Information based adaptive robotic exploration. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2002. 11, 20, 190, 191, 197

X. Bresson, S. Esedoglu, P. Vandergheynst, J.-P. Thiran, and S. Osher. Fast global minimization of the active contour/snake model. *Journal of Mathematical Imaging and Vision*, 28(2), 2007. 138, 157

M. Bryson and S. Sukkarieh. Observability analysis and active control for airborne SLAM. *IEEE Transactions on Aerospace and Electronic Systems*, 44(1), 2008. 190

M. Bryson, M. Johnson-Roberson, and S. Sukkarieh. Airborne smoothing and mapping using vision and inertial sensors. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3143–3148, 2009. 88

M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. Achtelik, and R. Siegwart. The EuRoC MAV datasets. *Int. J. of Robotics Research*, 2015. URL http://projects.asl.ethz.ch/datasets/doku.php?id=kmavvisualinertialdatasets. 15, 66, 67

L. Carlone, Z. Kira, C. Beall, V. Indelman, and F. Dellaert. Eliminating conditionally independent sets in factor graphs: A unifying perspective based on smart factors. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014. 85

A. Censi. Generalized monotone co-design problems; or, everything is the same. Technical report, Laboratory for Information and Decision Systems, MIT, 2015. 24

A. Censi, E. Mueller, E. Frazzoli, and S. Soatto. A power-performance approach to comparing sensor families, with application to comparing neuromorphic to traditional vision sensors. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015. 24

A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1), 2011. 135, 138, 158

S. Chen, Y. Li, and Ngai M. Kwok. Active vision in robotic systems: A survey of recent developments. *Int. J. of Robotics Research*, 30(11):1343–1377, 2011. 189

G. S. Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications (Applied and Numerical Harmonic Analysis)*. Birkhauser, 2012. 80, 89, 91

A. Chiuso, P. Favaro, H. Jin, and S. Soatto. Structure from motion causally integrated over time. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(4):523–535, Apr 2002. 3, 46

W. Churchill and P. Newman. Experience-based navigation for long-term localisation. *Int. J. of Robotics Research*, 2013. 24

T. Cieslewski, L. Simon, M. Dymczyk, S. Magnenat, and R. Siegwart. Map API - scalable decentralized map building for robots. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015. 24

J. Civera, A.J. Davison, and J. Montiel. Inverse depth parametrization for monocular slam. *IEEE Trans. Robotics*, 24(5), 2008. 57, 140

M. Cognetti, P. Stegagno, A. Franchi, G. Oriolo, and H. H. Buelthoff. 3D mutual localization with anonymous bearing measurements. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012. 29

A. I. Comport, E. Marchand, and F. Chaumette. A real-time tracker for markerless augmented reality. In *IEEE and ACM Int. Sym. on Mixed and Augmented Reality (ISMAR)*, 2003. 55

A.I. Comport, E. Malis, and P. Rives. Real-time quadrifocal visual odometry. *Int. J. of Robotics Research*, 29(2-3):245–266, January 2010. ISSN 0278-3649. doi: 10.1177/0278364909356601. 49

J. L. Crassidis. Sigma-point Kalman filtering for integrated GPS and inertial navigation. *IEEE Trans. Aerosp. Electron. Syst.*, 42(2):750–756, 2006. 99, 105

P. E. Crouch and R. Grossman. Numerical integration of ordinary differential equations on manifolds. *J. Nonlinear Sci.*, 3:1–22, 1993. 98

A. Cunningham, V. Indelman, and F. Dellaert. DDF-SAM 2.0: Consistent distributed smoothing and mapping. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013. 30

Z. Danping and T. Ping. CoSLAM: Collaborative visual slam in dynamic environments. *IEEE Trans. Pattern Anal. Machine Intell.*, 2012. 30

A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Int. Conf. on Computer Vision (ICCV)*, pages 1403–1410, 2003. 3, 133

A. J. Davison and R. M. Murray. Simultaneous localization and map-building using active vision. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(7), 2002. 12, 20, 190, 191

A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal. Machine Intell.*, 29(6):1052–1067, June 2007. 46, 87

C. de Boor. *A practical guide to splines*. Springer Verlag New York, 2001. 199

T. Delbruck, B. Linares-Barranco, E. Culurciello, and C. Posch. Activity-driven, event-based vision sensors. In *IEEE Intl. Symp. on Circuits and Systems (ISCAS)*, pages 2426–2429, May 2010. 23

F. Dellaert. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. Technical Report GIT-GVU-05-11, Georgia Institute of Technology, 2005. 96

F. Dellaert. Factor graphs and GTSAM: A hands-on introduction. Technical Report GT-RIM-CP&R-2012-002, Georgia Institute of Technology, September 2012. 86, 121

F. Dellaert and R. Collins. Fast image-based tracking by selective pixel integration. In *ICCV Workshop on Frame-Rate Vision*, 1999. 50, 66

F. Dellaert and M. Kaess. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *Int. J. of Robotics Research*, 25(12):1181–1203, December 2006. 8, 47, 76

V. Desaraju, N. Michael, M. Humenberger, R. Brockers, S. Weiss, and L. Matthies. Vision-based landing site evaluation and trajectory generation toward rooftop landing. In *Robotics: Science and Systems (RSS)*, 2014. 151, 152, 153

J. Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors. Technical report, Stanford University, 2006. 129

M. Ding, K. Lyngbaek, and A. Zakhor. Automatic registration of aerial imagery with untextured 3d lidar models. *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, pages 1–8, 2008. 172

T-C. Dong-Si and A.I. Mourikis. Motion tracking with fixed-lag smoothing: Algorithm consistency and analysis. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011. 88

M. Donoser and H. Bischof. Efficient maximally stable extremal region (MSER) tracking. *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 1:553–560, 2006. ISSN 1063-6919. doi: 10.1109/CVPR.2006.107. 172

T. Drummond and R. Cipolla. Real-time visual tracking of complex structures. *IEEE Trans. Pattern Anal. Machine Intell.*, 24:932–946, 2002. 55

H. F. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (SLAM): Part I. *IEEE Robotics Automation Magazine*, 13(2):99–110, June 2006. ISSN 1070-9932. doi: 10.1109/MRA.2006.1638022. 2

M. Dymczyk, S. Lynen, T. Cieslewski, M. Bosse, R. Siegwart, and P. Furgale. The gist of maps - summarizing experience for lifelong localization. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015. 24

F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard. An evaluation of the RGB-D SLAM system. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012. 70

J. Engel, J. Sturm, and D. Cremers. Semi-dense visual odometry for a monocular camera. In *Int. Conf. on Computer Vision (ICCV)*, 2013. 50, 53, 59, 66, 70

J. Engel, J. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *Eur. Conf. on Computer Vision (ECCV)*, 2014. 7, 8, 47, 50, 54, 67, 70, 71

N. Engelhard, F. Endres, J. Hess, J. Sturm, and W. Burgard. Real-time 3d visual slam with a hand-held RGB-D camera. *Proc. RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum*, April 2011. 174

C. Evans. Notes on the OpenSURF library. Technical report, University of Bristol, 2009. 35

M. Faessler, F. Fontana, C. Forster, E. Mueggler, M. Pizzoli, and D. Scaramuzza. Autonomous, vision-based flight and live dense 3D mapping with a quadrotor MAV. *J. of Field Robotics*, pages 1556–4967, 2015. URL http://dx.doi.org/10.1002/rob.21581. 6, 56, 70, 76, 77, 151, 152, 163

P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart. Robot-centric elevation mapping with uncertainty estimates. In *Int. Conf. on Climbing and Walking Robots (CLAWAR)*, 2014. 152, 154, 158, 159, 160, 161

J. A. Farrell. *Aided Navigation: GPS with High Rate Sensors*. McGraw-Hill, 2008. 94, 98

O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT University Press, 1994. 4

H. S. S. Feder, J. J. Leonard, and C. M. Smith. Adaptive Mobile Robot Navigation and Mapping. *Int. J. of Robotics Research*, 18(7):650–558, 1999. 190, 197

M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. ISSN 0001-0782. doi: http://doi.acm.org/10.1145/358669.358692. 36, 49, 77

C. Forster, M. Pizzoli, and D. Scaramuzza. Air-ground localization and map augmentation using monocular dense reconstruction. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3971–3978, 2013. URL http://dx.doi.org/10.1109/IROS.2013.6696924. 11, 134, 136

C. Forster, M. Pizzoli, and C. Scaramuzza. Appearance-based active, monocular, dense depth estimation for micro aerial vehicles. In *Robotics: Science and Systems (RSS)*, 2014a. URL http://www.roboticsproceedings.org/rss10/p29.html. 11, 12, 19, 153, 154, 163

C. Forster, M. Pizzoli, and D. Scaramuzza. SVO: Fast semi-direct monocular visual odometry. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 15–22, 2014b. URL http://dx.doi.org/10.1109/ICRA.2014.6906584. 9, 11, 17, 48, 56, 115, 140, 152, 154, 192, 202

C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. On-manifold preintegration theory for fast and accurate visual-inertial navigation. December 2015a. URL http://arxiv.org/pdf/1512.02363v1.pdf. 50, 56, 73, 76, 78

C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. In *Robotics: Science and Systems (RSS)*, 2015b. URL http://www.roboticsproceedings.org/rss11/p06.html. 13, 16, 86

C. Forster, M. Faessler, F. Fontana, M. Werlberger, and D. Scaramuzza. Continuous on-board monocular-vision–based aerial elevation mapping for quadrotor landing. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 111–118, 2015c. URL http://dx.doi.org/10.1109/ICRA.2015.7138988. 11, 18, 70

D. Fox, W. Burgard, H. Kruppa, and S. Thrun. A probabilistic approach to collaborative multi-robot localization. *IEEE Aerospace Conf.*, 8(3):325–344, 2000. 29

J.-M. Frahm, P. Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building rome on a cloudless day. In *Eur. Conf. on Computer Vision (ECCV)*, 2010. 3

S. Fuhrmann and M. Goesele. Floating scale surface reconstruction. In *SIGGRAPH*, volume 33, pages 1–11. ACM, July 2014. ISBN 9781450329040. doi: 10.1145/2601097.2601163. 10

P. Furgale, J. Rehder, and R. Siegwart. Unified temporal and spatial calibration for multi-sensor systems. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2013. 51, 61, 99, 116

Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. Pattern Anal. Machine Intell.*, 32(8):1362–1376, 2010. 9, 133, 175

D. Gallup, J.-M. Frahm, P. Mordohai, Qingxiong Yang, and M. Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2007. 10, 134, 152

P. J. Garcia-Pardo, G. S. Sukhatme, and J. F. Montgomery. Towards vision-based safe landing for an autonomous helicopter. *Journal of Robotics and Autonomous Systems*, 38, 2002. 152

A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2012. 116, 117

A. J. Glover, W. P. Maddern, M. J. Milford, and G. F. Wyeth. FAB-MAP + RatSLAM: appearance-based slam for multiple times of day. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2010. 35

B. Goldlücke, M. Aubry, K. Kolev, and D. Cremers. A superresolution framework for high-accuracy multiview reconstruction. In *31st DAGM Symposium. Proceedings*, 2009. doi: 10.1007/978-3-642-03798-6_35. 9

Google. Google Self-Driving Car Project, Monthly Report. https://static.googleusercontent.com/media/www.google.com/en//selfdrivingcar/files/reports/report-0515.pdf, 2015. [Online; accessed 30-November-2015]. 2

V. Grabe, H. H. Bülthoff, and P. Robuffo Giordano. A comparison of scale estimation schemes for a quadrotor UAV based on optical flow and IMU measurements. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2013. 6

G. Graber, T. Pock, and H. Bischof. Online 3d reconstruction using convex optimization. In *Int. Conf. on Computer Vision Workshops (ICCV Workshops)*, pages 708–711. IEEE, November 2011. ISBN 978-1-4673-0063-6. doi: 10.1109/ICCVW.2011.6130318. 134

L. Gruber, S. Gauglitz, J. Ventura, S. Zollmann, M. Huber, M. Schlegel, G. Klinker, D. Schmalstieg, and D. Höllerer. The city of sights: Design, construction, and measurement of an augmented reality stage set. In *IEEE and ACM Int. Sym. on Mixed and Augmented Reality (ISMAR)*, 2010. 143, 145

A. Handa, R. A. Newcombe, A. Angeli, and A. J. Davison. Real-time camera tracking: When is high frame-rate best? In *Eur. Conf. on Computer Vision (ECCV)*, 2012. 7, 141

A. Handa, T. Whelan, J.B. McDonald, and A.J. Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Hong Kong, China, May 2014. 15, 49, 67, 71, 72, 73

C. Häne, C. Zach, A. Cohen, R. Angst, and M. Pollefeys. Joint 3D scene reconstruction and class segmentation. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2013. 24

C. Harris and C. Stennett. RAPiD - a video-rate object tracker. In *British Machine Vision Conf. (BMVC)*, pages 73–78, 1990. 55

C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of The Fourth Alvey Vision Conference*, volume 15, pages 147–151. Manchester, UK, 1988. URL http://www.cis.rit.edu/~cnspci/references/dip/harris1988.pdf. 4

R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. Second Edition. 133

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. 4, 106

J. Hernandez, K. Tsotsos, and S. Soatto. Observability, identifiability and sensitivity of vision-aided inertial navigation. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015. 88

J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis. Camera-imu-based localization: Observability analysis and consistency improvement. *Int. J. of Robotics Research*, 33(1):182–201, 2014. 88, 111

H. Hirschmuller and D. Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *IEEE Trans. Pattern Anal. Machine Intell.*, 31(9), 2009. 134

J. Hornegger. Statistical modeling of relations for 3-D object recognition. In *Intl. Conf. Acoust., Speech, and Signal Proc. (ICASSP)*, volume 4, pages 3173–3176, Munich, April 1997. 113

J. Hornegger and C. Tomasi. Representation issues in the ML estimation of camera motion. In *Int. Conf. on Computer Vision (ICCV)*, pages 640–647, Kerkyra, Greece, September 1999. 89

A. Howard. Multi-robot simultaneous localization and mapping using particle filters. *Int. J. of Robotics Research*, 25(12):1243–1256, 2006. 29

M. A. Hsieh, A. Cowley, J. F. Keller, L. Chaimowicz, B. Grocholsky, V. Kumar, C. J. Taylor, Y. Endo, R. C. Arkin, B. Jung, D. F. Wolf, G. S. Sukhatme, and D. C. MacKenzie. Adaptive teams of autonomous aerial and ground robots for situational awareness. *J. of Field Robotics*, 24(11–12):991–1014, 2007. doi: 10.1002/rob.20222. 171

A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy. Visual odometry and mapping for autonomous flight using an RGB-D camera. In *Proc. of the Int. Symp. of Robotics Research (ISRR)*, Flagstaff, USA, August 2011a. 72

G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis. A first-estimates jacobian EKF for improving SLAM consistency. In *Int. Sym. on Experimental Robotics (ISER)*, 2008. 87

G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis. Observability-based rules for designing consistent EKF SLAM estimators. *Int. J. of Robotics Research*, 29:502–528, May 2010. 8, 47

G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis. An observability-constrained sliding window filter for SLAM. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 65–72, 2011b. 88

S. Huang, N. M. Kwok, G. Dissanayake, Q. P. Ha, and G. Fang. Multi-Step Look-Ahead Trajectory Planning in SLAM: Possibility and Necessity. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2005. 198, 199

T. Huang and O. Faugeras. Some properties of the E matrix in two-view motion estimation. *IEEE Trans. Pattern Anal. Machine Intell.*, 11(12):1310–1312, 1987. 3

X. Huang, I. Walker, and S. Birchfield. Occlusion-Aware Reconstruction and Manipulation of 3D Articulated Objects. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012. 189

V. Indelman, S. Wiliams, M. Kaess, and F. Dellaert. Factor graph based incremental smoothing in inertial navigation systems. In *Int. Conf. on Information Fusion (FUSION)*, 2012. 88, 99

V. Indelman, A. Melim, and F. Dellaert. Incremental light bundle adjustment for robotics navigation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, November 2013a. 89

V. Indelman, S. Wiliams, M. Kaess, and F. Dellaert. Information fusion in navigation systems via factor graph based incremental smoothing. *Journal of Robotics and Autonomous Systems*, 61 (8):721–738, August 2013b. 9, 88, 89

M. Irani and P. Anandan. All about direct methods. In *Proc. Workshop Vis. Algorithms: Theory Pract.*, pages 267–277, 1999. 7, 14, 47, 49, 190

S. Izadi, R. A. Newcombe, D. Kim, O. Hilliges, D. Molyneaux, S. Hodges, P. Kohli, J. Shotton, A.J. Davison, and A. Fitzgibbon. KinectFusion: Real-time dynamic 3D surface reconstruction and interaction. In *SIGGRAPH*, page 23, August 2011. 176

H. Jin, P. Favaro, and S. Soatto. Real-time feature tracking and outlier rejection with changes in illumination. *Int. Conf. on Computer Vision (ICCV)*, 1, 2001. doi: 10.1109/ICCV.2001.937588. 63

H. Jin, P. Favaro, and S. Soatto. A semi-direct approach to structure from motion. *The Visual Computer*, 19(6):377–394, 2003. 49

A. Johnson, A. Klumpp, J. Collier, and A. Wolf. Lidar-based hazard avoidance for safe landing on mars. *AIAA Jour. Guidance, Control and Dynamics*, 25(5), 2002. 5, 151

A. Johnson, J. Montgomery, and L. Matthies. Vision guided landing of an autonomous helicopter in hazardous terrain. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2005. 151, 152

E.S. Jones and S. Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *Int. J. of Robotics Research*, 30(4), Apr 2011. 87

S-H. Jung and C.J. Taylor. Camera trajectory estimation using inertial sensor measurements and structure fom motion results. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2001. 9, 88

M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Trans. Robotics*, 24(6):1365–1378, December 2008. 88

M. Kaess, V. Ila, R. Roberts, and F. Dellaert. The Bayes tree: Enabling incremental reordering and fluid relinearization for online mapping. Technical Report MIT-CSAIL-TR-2010-021, Computer Science and Artificial Intelligence Laboratory, MIT, January 2010. 108

M. Kaess, H. Johannsson, R. Roberts, V. Ila, J.J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *Int. J. of Robotics Research*, 31:217–236, February 2012. 47, 56, 67, 76, 85, 87, 88, 108, 109, 115

S.B. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2001. 133

N. Keivan, A. Patron-Perez, and G. Sibley. Asynchronous adaptive conditioning for visual-inertial SLAM. In *Int. Sym. on Experimental Robotics (ISER)*, 2014. 89

C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for rgb-d cameras. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013. 49, 62, 70, 72

G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *IEEE and ACM Int. Sym. on Mixed and Augmented Reality (ISMAR)*, pages 225–234, Nara, Japan, November 2007. 31, 34, 48, 50, 54, 70, 88, 174

G. Klein and D. Murray. Improving the agility of keyframe-based SLAM. In *Eur. Conf. on Computer Vision (ECCV)*, pages 802–815, 2008. 7, 47, 55

G. Klein and D. Murray. Parallel tracking and mapping on a camera phone. In *IEEE and ACM Int. Sym. on Mixed and Augmented Reality (ISMAR)*, 2009. 88

L. Kneip, M. Chli, and R. Siegwart. Robust real-time visual odometry with a single camera and an IMU. In *British Machine Vision Conf. (BMVC)*, 2011a. 31, 174

L. Kneip, D. Scaramuzza, and R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, pages 2969–2976, 2011b. 36

D. G. Kottas, J. A. Hesch, S. L. Bowman, and S. I. Roumeliotis. On the consistency of vision-aided inertial navigation. In *Int. Sym. on Experimental Robotics (ISER)*, 2012. 50, 87, 88, 109, 111

S. Kriegel, T. Bodenmüller, M. Suppa, and G. Hirzinger. A surface-based next-best-view approach for automated 3d model completion of unknown objects. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011. 11, 20

S. Kriegel, C. Rink, T. Bodenmüller, and M. Suppa. Efficient next-best-scan planning for autonomous 3d surface reconstruction of unknown objects. *J. of Real-Time Image Processing*, pages 1–21, 2013. 189, 191

F.R. Kschischang, B.J. Frey, and H-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. Inform. Theory*, 47(2), February 2001. 96

R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Shanghai, China, May 2011. 8, 33, 35, 37, 47

K.Y.K. Leung, C.M. Clark, and J.P. Huissoon. Localization in urban environments by matching ground level video images with an aerial image. *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 551 –556, May 2008. doi: 10.1109/ROBOT.2008.4543264. 176

S. Leutenegger, M. Chli, and R.Y. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *Int. Conf. on Computer Vision (ICCV)*, pages 2548–2555, November 2011. doi: 10.1109/ICCV.2011.6126542. 4, 35, 40

S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart. Keyframe-based visual-inertial slam using nonlinear optimization. In *Robotics: Science and Systems (RSS)*, 2013. 88, 94, 120

S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-based visual-inertial slam using nonlinear optimization. *Int. J. of Robotics Research*, 2015. 88, 89, 100, 116, 117, 119

M. Li and A.I. Mourikis. Online temporal calibration for camera-imu systems: Theory and algorithms. *Int. J. of Robotics Research*, 33(6), 2014. 99

L. Liu and I. Stamos. Multiview geometry for texture mapping 2d images onto 3d range data. *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, pages 2293–2300, 2006. 172

H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, (293):133–135, 1981. 3

S. Lovegrove, A. J. Davison, and J. Ibañez Guzmán. Accurate visual odometry from a rear parking camera. *IEEE Intelligent Vehicles Symposium, Proceedings*, pages 788–793, 2011. ISSN 1931-0587. doi: 10.1109/IVS.2011.5940546. 7, 47

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60(2): 91–110, November 2004. ISSN 0920-5691. 4

B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Int. Joint Conf. on Artificial Intelligence*, pages 121–130, 1981. 77

T. Lupton and S. Sukkarieh. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. *IEEE Trans. Robotics*, 28(1):61–76, February 2012. 9, 16, 85, 89, 99, 100, 104, 108, 113, 121

S. Lynen, M. Achtelik, S. Weiss, M. Chli, and R. Siegwart. A robust and modular multi-sensor fusion approach applied to MAV navigation. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2013. 154

Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer Verlag, 2005. 4, 52

K. MacTavish and T. D. Barfoot. At all costs: A comparison of robust cost functions for camera correspondence outliers. In *Conf. on Computer and Robot Vision (CRV)*, 2015. 49, 62

M. Maimone, Y. Cheng, and L. Matthies. Two years of visual odometry on the mars exploration rovers. *J. of Field Robotics*, 24(3):169–186, 2007. ISSN 1556-4967. doi: 10.1002/rob.20184. URL http://dx.doi.org/10.1002/rob.20184. 3, 7, 8, 47

J. H. Manton. Optimization algorithms exploiting unitary constraints. *IEEE Trans. Signal Processing*, 50:63–650, 2002. 94

A. Martinelli. Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. *IEEE Trans. Robotics*, 28(1):44–60, February 2012. ISSN 1552-3098. doi: 10.1109/TRO.2011.2160468. 85, 88

A. Martinelli. Observability properties and deterministic algorithms in visual-inertial structure from motion. *Foundations and Trends in Robotics*, pages 1–75, 2013. 16, 87, 88

A. Martinelli, F. Pont, and R. Siegwart. Multi-robot localization using relative observations. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2005. 29

L. Matthies, R. Szeliski, and R. Kanade. Incremental estimation of dense depth maps from image sequences. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 1988. 133, 193

P. Maybeck. *Stochastic Models, Estimation and Control*, volume 1. Academic Press, New York, 1979. 88

J. McDonald, M. Kaess, C. Cadena, J. Neira, and J. J. Leonard. 6-DOF Multi-session Visual SLAM using Anchor Nodes. *European Conf. on Mobile Robots (ECMR)*, 2011. 30

C. Mei, S. Benhimane, E. Malis, and P. Rives. Efficient homography-based tracking and 3-d reconstruction for single-viewpoint sensors. *IEEE Trans. Robotics*, 24(6):1352–1364, December 2008. ISSN 1552-3098. doi: 10.1109/TRO.2008.2007941. 49

M. Meilland and A.I. Comport. On unifying key-frame and voxel-based dense visual SLAM at large scales. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, 3-8 November 2013. IEEE/RSJ. 49, 134

M. Meilland, A. Comport, and P. Rives. Real-time dense visual tracking under large lighting variations. In *British Machine Vision Conf. (BMVC)*, 2011. ISBN 1-901725-43-X. doi: 10.5244/C.25.45. 49

C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000. 128

N. Michael, S. Shen, K. Mohta, Y. Mulgaonkar, V. Kumar, K. Nagatani, Y. Okada, S. Kiribayashi, K. Otake, K. Yoshida, K. Ohno, E. Takeuchi, and S. Tadokoro. Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *J. of Field Robotics*, 29(5):832–841, September 2012. 169, 171

M. Moakher. Means and averaging in the group of rotations. *SIAM Journal on Matrix Analysis and Applications*, 24(1):1–16, 2002. 89

N. D. Molton, A. J. Davison, and I. D. Reid. Locally planar patch features for real-time structure from motion. In *British Machine Vision Conf. (BMVC)*. BMVC, September 2004. 49

H. P. Moravec. *Obstacle Avoidance and Navigation in the Real World by Seeing Robot Rover*. PhD thesis, Carnegie-Mellon University, Pittsburgh, Pennsylvania, September 1980. 3

J.-M. Morel and G. Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, April 2009. ISSN 1936-4954. 171, 172

E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. 3D reconstruction of complex structures with bundle adjustment: an incremental approach. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2006. 35

A. I. Mourikis and S. I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3565–3572, April 2007. 85, 86, 87, 107, 116, 117, 120

A. I. Mourikis and S. I. Roumeliotis. A dual-layer estimator architecture for long-term localization. In *Proc. of the Workshop on Visual Localization for Mobile Platforms at CVPR*, Anchorage, Alaska, June 2008. 88

E. Mueggler, G. Gallego, and D. Scaramuzza. Continuous-time trajectory estimation for event-based vision sensors. In *Robotics: Science and Systems (RSS)*, 2015. 23

H. Munthe-Kaas. Higher order runge-kutta methods on manifolds. *Appl. Numer. Math.*, 29(1): 115–127, 1999. 98

R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Trans. Robotics*, 31(5):1147–1163, 2015a. 48, 67, 70, 72, 75

Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. `arXiv:1502.00956`, February 2015b. 69

R. M. Murray, Z. Li, and S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994. 98

E.D. Nerurkar, K.J. Wu, and S.I. Roumeliotis. C-KLAM: Constrained keyframe-based localization and mapping. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014. 88, 94

R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *IEEE and ACM Int. Sym. on Mixed and Augmented Reality (ISMAR)*, pages 127–136, Basel, Switzerland, October 2011a. 72

R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. DTAM: Dense tracking and mapping in real-time. In *Int. Conf. on Computer Vision (ICCV)*, pages 2320–2327, November 2011b. 7, 8, 10, 24, 47, 49, 50, 53, 66, 134, 135, 138, 152, 170, 175, 188, 189, 197

R. A. Newcombe, D. Fox, and S. Seitz. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2015. 24

J. Nikolic, J. Rehder, M. Burri, P. Gohl, S. Leutenegger, P. Furgale, and R. Siegwart. A synchronized visual-inertial sensor system with FPGA pre-processing for accurate real-time SLAM. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014. 67, 116

N. Nilsson. *The Quest for Artificial Intelligence*. Cambridge University Press, 2009. 1

D. Nister. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Machine Intell.*, 26(6):756–777, 2004. 62

D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 652–659, June 2004. 46

G. Nuetzi, S. Weiss, D. Scaramuzza, and R. Siegwart. Fusion of IMU and vision for absolute scale estimation in monocular slam. *J. of Intelligent and Robotic Systems*, 61:287–299, 2011. 173

P. Ondruska, P. Kohli, and S. Izadi. Mobilefusion: Real-time volumetric surface reconstruction and dense tracking on mobile phones. In *IEEE and ACM Int. Sym. on Mixed and Augmented Reality (ISMAR)*, Fukuoka, Japan, October 2015. 8, 50

L. Pack Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *J. Artificial Intell.*, 101, 1998. 199

F. C. Park and B. J. Martin. Robot sensor calibration: Solving AX=XB on the euclidean group. *IEEE Trans. Robotics*, 10(5), 1994. 116

J. Park and W.-K. Chung. Geometric integration on euclidean group with application to articulated multibody systems. *IEEE Trans. Robotics*, 2005. 98

A. Patron-Perez, S. Lovegrove, and G. Sibley. A spline-based trajectory representation for sensor fusion and rolling shutter cameras. *Int. J. Comput. Vis.*, February 2015. 88

M. Pizzoli, C. Forster, and D. Scaramuzza. REMODE: Probabilistic, monocular dense reconstruction in real time. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2609–2616, 2014. URL http://dx.doi.org/10.1109/ICRA.2014.6907233. 11, 18, 49, 57, 59, 69, 151, 152, 154, 155, 157, 158, 189, 192, 195

M. Pollefeys, R. Koch, M. Vergauwen, and L. Van Gool. Hand-held acquisition of 3D models with a video camera. In *Proceedings of the IEEE International Workshop on 3D Digital Imaging and Modeling (3DIM)*, 1999. 5

M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *Int. J. Comput. Vis.*, 59:207–232, 2004. 5

M. Pollefeys, D. Nister, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewenius, R. Yang, G. Welcha, and H. Towles. Detailed real-time urban 3D reconstruction from video. *Int. J. Comput. Vis.*, 78(2–3):143–167, 2008. 4, 5

F. Pomerleau, S. Magnenat, F. Colas, M. Liu, and R. Siegwart. Tracking a depth camera: Parameter exploration for fast icp. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3824–3829, 2011. 174, 180

A. Pretto, E. Menegatti, and E. Pagello. Omnidirectional dense large-scale mapping and navigation based on meaningful triangulation. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3289–3296. IEEE, May 2011. ISBN 978-1-61284-386-5. doi: 10.1109/ICRA.2011.5980206. 49

G. Reitmayr and T.W. Drummond. Going out: robust model-based tracking for outdoor augmented reality. In *IEEE and ACM Int. Sym. on Mixed and Augmented Reality (ISMAR)*, pages 109–118, October 2006. 55

C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, pages 3017–3024, 2011. 170, 175

R. Rocha, J. Dias, and A. Carvalho. Cooperative multi-robot systems: A study of vision-based 3-d mapping using information theory. *Journal of Robotics and Autonomous Systems*, 2005. 29

E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Anal. Machine Intell.*, 32(1):105–119, January 2010. ISSN 0162-8828. doi: 10.1109/TPAMI.2008.275. 63

S. I. Roumeliotis and J. W. Burdick. Stochastic cloning: A generalized framework for processing relative state measurements. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2002. 87

L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1–4), 1992. 10, 134

P. Rudol, M. Wzorek, G. Conte, and P. Doherty. Micro unmanned aerial vehicle visual servoing for cooperative indoor exploration. In *IEEE Trans. on Automatic Control*, pages 1–10, 2008. 170

R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Shanghai, China, May 2011. 176

R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison. SLAM++: Simultaneous localisation and mapping at the level of objects. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2013. 24

S. Saripalli, J. F. Montgomery, and G. S. Sukhatme. Vision-based autonomous landing of an unmanned aerial vehicle. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2002. 152

D. Scaramuzza. 1-point-RANSAC structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. *Int. J. Comput. Vis.*, 96, 2011. 4

D. Scaramuzza and F. Fraundorfer. Visual odometry [tutorial]. Part I: The first 30 years and fundamentals. *IEEE Robotics Automation Magazine*, 18(4):80 –92, December 2011. ISSN 1070-9932. doi: 10.1109/MRA.2011.943233. 46, 48

D. Scaramuzza, A. Martinelli, and R. Siegwart. A flexible technique for accurate omnidirectional camera calibration and structure from motion. In *Int. Conf. on Computer Vision Systems (ICVS)*, pages 45–45, 2006. 60

D. Scaramuzza, M.C. Achtelik, L. Doitsidis, F. Fraundorfer, E. B. Kosmatopoulos, A. Martinelli, M. W. Achtelik, M. Chli, S.A. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, G.H. Lee, S. Lynen, L. Meier, M. Pollefeys, A. Renzaglia, R. Siegwart, J. C. Stumpf, P. Tanskanen, C. Troiani, and S. Weiss. Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in GPS-denied environments. *IEEE Robotics Automation Magazine*, 2014. 6, 39, 151

S. Scherer, L. J. Chamberlain, and S. Singh. Autonomous landing at unprepared sites by a full-scale helicopter. *Journal of Robotics and Autonomous Systems*, 2012. 5, 151

K. Schmid, H. Hirschmüller, A. Dömel, I. Grixa, M. Suppa, and G. Hirzinger. View planning for multi-view stereo 3D reconstruction using an autonomous multicopter. *J. of Intelligent and Robotic Systems*, 65(1-4):309–323, 2012. 189

W. Scott, G. Roth, and J.-F. Rivest. View planning for automated 3d object reconstruction inspection. *ACM Computing Surveys*, 35(1), 2003. 189

S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2006. 9, 133

S. Shen. *Autonomous Navigation in Complex Indoor and Outdoor Environments with Micro Aerial Vehicles*. PhD thesis, University of Pennsylvania, Philadelphia, PA, USA, July 2014. 89

S. Shen, N. Michael, and V. Kumar. Autonomous indoor 3D exploration with a micro-aerial vehicle. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012. 6

G. Sibley, L. Matthies, and G. Sukhatme. Sliding window filter with application to planetary landing. *J. of Field Robotics*, 27(5):587–608, 2010. 88

G. Silveira, E. Malis, and P. Rives. An efficient direct approach to visual slam. *IEEE Trans. Robotics*, 2008. 49

R. Sim and N. Roy. Global a-optimal robot exploration in SLAM. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2005. 12, 20, 191

J. Sivic and A. Zisserman. Video Google: a text retrieval approach to object matching in videos. In *Int. Conf. on Computer Vision (ICCV)*, 2003. doi: 10.1109/ICCV.2003.1238663. 35

S. T. Smith. Optimization techniques on Riemannian manifolds. *Hamiltonian and Gradient Flows, Algorithms and Control, Fields Inst. Commun., Amer. Math. Soc.*, 3:113–136, 1994. 93

S. Soatto. Actionable information in vision. In *Int. Conf. on Computer Vision (ICCV)*, 2009. 12, 20, 25, 190, 191, 197

S. Soatto. Steps Towards a Theory of Visual Information: Active Perception, Signal-to-Symbol Conversion and the Interplay Between Sensing and Control. *ArXiv e-prints*, 2011. 190, 198

J. Sola, A. Monin, M. Devy, and T. A. Vidal-Calleja. Fusing monocular information in multicamera SLAM. In *Robotics: Science and Systems (RSS)*, 2008. 29

C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using Rao-Blackwellized particle filters. In *Robotics: Science and Systems (RSS)*, 2005. 12, 20, 190, 191

J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 2012. ISSN 0893-6080. doi: 10.1016/j.neunet.2012.02.016. 2

T. Stentz, A. Kelly, H. Herman, P. Rander, O. Amidi, and R. Mandelbaum. Integrated air/ground vehicle system for semi-autonomous off-road navigation. *Robotics Institute*, 2002. 171

D. Sterlow and S. Singh. Motion estimation from image and inertial measurements. *Int. J. of Robotics Research*, 2004. 9, 88

H. Strasdat, J.M.M. Montiel, and A.J. Davison. Real-time monocular SLAM: Why filter? In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2010. 34, 37, 88, 95

H. Strasdat, A.J. Davison, J.M.M. Montiel, and K. Konolige. Double window optimisation for constant time visual SLAM. In *Int. Conf. on Computer Vision (ICCV)*, 2011. 174

C. Strecha, W. von Hansen, L. Van Gool, Pascal Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, pages 1–8. IEEE, June 2008. ISBN 978-1-4244-2242-5. doi: 10.1109/CVPR.2008.4587706. 9

J. Stühmer, S. Gumhold, and D. Cremers. Real-time dense geometry from a handheld camera. In *Pattern Recognition*, volume 6376 of *Lecture Notes in Computer Science*, pages 11–20, 2010. ISBN 978-3-642-15985-5. doi: 10.1007/978-3-642-15986-2_2. 10, 134, 152, 188

J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Oct. 2012. 67, 68, 70, 71

R. Szeliski and D. Scharstein. Sampling the disparity space image. *IEEE Trans. Pattern Anal. Machine Intell.*, 26(3):419–425, 2004. 134

S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, Cambridge, MA, 2005. 6, 178

E. Tola, C. Strecha, and P. Fua. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *Machine Vision and Applications*, 23(5):903–920, May 2012. ISSN 09328092. doi: 10.1007/s00138-011-0346-8. 9

C. Tomasi and T. Kanade. Shape and motion from image streams: a factorization method. *Int. J. Comput. Vis.*, (7597):137–154, 1992. 46

N. Trawny, S.I. Roumeliotis, and G.B. Giannakis. Cooperative multi-robot localization under communication constraints. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2009. 29

B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *LNCS*, pages 298–372. Springer Verlag, 2000. 8, 14, 47, 48

K. Tsotsos, A. Chiuso, and S. Soatto. Robust inference for visual-inertial sensor fusion. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015. 87

T. Tykkala, C. Audras, and A.I. Comport. Direct iterative closest point for real-time visual odometry. In *Int. Conf. on Computer Vision (ICCV)*, 2011. 49

S. Ullman. *The Interpretation of Visual Motion*. MIT Press: Cambridge, MA, 1979. 46

S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Machine Intell.*, 13(4), 1991. 68

L. Vacchetti, V. Lepetit, and P. Fua. Combining edge and texture information for real-time accurate 3d camera tracking. In *IEEE and ACM Int. Sym. on Mixed and Augmented Reality (ISMAR)*, 2004. 55

R. Valencia, J. Valls Miró, G. Dissanayake, and J. Andrade-Cetto. Active pose SLAM. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012. 12, 20, 190, 191

T. A. Vidal-Calleja, A. Sanfeliu, and J. Andrade-Cetto. Action Selection for Single-Camera SLAM. *IEEE Transactions on Systems, Man, and Cybernetics*, 40(6), 2010. 190

T. A. Vidal-Calleja, C. Berger, J. Solà, and S. Lacroix. Large scale multiple robot visual mapping with heterogeneous landmarks in semi-structured terrain. *Journal of Robotics and Autonomous Systems*, 59(9):654–674, September 2011. ISSN 09218890. 29, 171

G. Vogiatzis and C. Hernández. Video-based, real-time multi view stereo. *Image Vision Comput.*, 29(7):434–441, 2011. 10, 11, 17, 57, 58, 59, 135, 137, 140, 143, 147, 148, 156, 157, 188, 192

Y. Wang and G.S. Chirikjian. Error propagation on the euclidean group with applications to manipulator kinematics. *IEEE Trans. Robotics*, 22(4):591–602, 2006. 92

Y. Wang and G.S. Chirikjian. Nonparametric second-order theory of error propagation on motion groups. *Int. J. of Robotics Research*, 27(11–12):1258–1273, 2008. 89, 92

S. Weiss, M. Achtelik, L. Kneip, D. Scaramuzza, and R. Siegwart. Intuitive 3D maps for MAV terrain exploration and obstacle avoidance. *J. of Intelligent and Robotic Systems*, 61:473–493, 2011a. 151, 152

S. Weiss, D. Scaramuzza, and R. Siegwart. Monocular-SLAM-based navigation for autonomous micro helicopters in GPS-denied environments. *J. of Field Robotics*, 28(6):854–874, November 2011b. 29, 170

A. Wendel, A. Irschara, and H. Bischof. Automatic alignment of 3d reconstructions using a digital surface model. In *Workshop IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 29–36, June 2011. 172, 177

A. Wendel, M. Maurer, G. Graber, T. Pock, and H. Bischof. Dense reconstruction on-the-fly. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2012. 151, 152, 195

M. Werlberger, T. Pock, and H. Bischof. Motion estimation with non-local total variation regularization. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2010. 10, 134

P. Whaite and F. P. Ferrie. Autonomous Exploration: Driven by Uncertainty. *IEEE Trans. Pattern Anal. Machine Intell.*, 19(3):193–205, 1997. 190

T. Whelan, H. Johannsson, M. Kaess, J. J. Leonard, and J. B. McDonald. Robust real-time visual odometry for dense RGB-D mapping. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013. 72, 134

T. Whelan, M. Kaess, H. Johannsson, M.F. Fallon, J.J. Leonard, and J.B. McDonald. Real-time large scale dense RGB-D SLAM with volumetric fusion. *Int. J. of Robotics Research*, 2014. 8, 49, 50

T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison. ElasticFusion: Dense SLAM without a pose graph. In *Robotics: Science and Systems (RSS)*, Rome, Italy, July 2015. 8, 49, 50

C. Wu, B. Clipp, X. Li, J.-M. Frahm, and M. Pollefeys. 3d model matching with viewpoint-invariant patches (VIP). *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, pages 1–8, 2008. ISSN 1063-6919. 172

K. J. Wu, A. M. Ahmed, G. A. Georgiou, and S. I. Roumeliotis. A square root inverse filter for efficient vision-aided inertial navigation on mobile devices. In *Robotics: Science and Systems (RSS)*, 2015. 87

C. Zach. Fast and high quality fusion of depth maps. In *Proc. of 3DPVT*, 2008. 134, 176

C. Zach, T. Pock, and H. Bischof. A globally optimal algorithm for robust TV-L1 range image integration. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 1–8, 2007. ISBN 978-1-4244-1631-8. doi: 10.1109/ICCV.2007.4408983. 9

Z. Zhang, H. Rebecq, C. Forster, and D Scaramuzza. Benefit of large field-of-view cameras for visual odometry. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2016. 64, 74

W. Zhao, D. Nister, and S. Hsu. Alignment of continuous video onto 3d point clouds. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(8):1305 –1318, August 2005. 172

S. Zingg, D. Scaramuzza, S. Weiss, and R. Siegwart. MAV navigation through indoor corridors using optical flow. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2010. 6

J.-C. Zufferey and D. Floreano. Fly-inspired visual steering of an ultralight indoor aircraft. *IEEE Trans. Robotics*, 22(1):137–146, February 2006. 6

CHRISTIAN FORSTER

PERSONAL DATA

Nationality Swiss
Date of Birth December 25, 1986
Contact Hermetschloostr. 27, 8048 Zürich, Switzerland
☎ +41 79 568 32 75
✉ forster@ifi.uzh.ch

EDUCATION

2012 – 2016 **Robotics and Perception Group, University of Zurich**

PhD Student Computer Science

Advisor: Prof. Davide Scaramuzza
My research interest is in computer vision algorithms for realtime 3D environment reconstruction with a single camera for robotic applications.

2010 – 2012 **Swiss Federal Institute of Technology (ETH), Zurich**

MSc ETH in Robotics, Systems and Control

Tutor: Prof. Roland Siegwart, Autonomous Systems Lab
Thesis: *Collaborative Structure from Motion*
Thesis advisors: Laurent Kneip, Simon Lynen, Margarita Chli
Graduation with distinction

2006 – 2009 **Swiss Federal Institute of Technology (ETH), Zurich**

BSc ETH in Mechanical Engineering

Area of specialization: Mechatronics
Thesis: *Distributed Coverage Control – Mussy Swarms or Neat Team of Agents?*
Thesis advisor: Andreas Breitenmoser, Autonomous Systems Lab

2001 – 2005 **Kantonsschule Zuercher Oberland, Zurich**

Matura

High school diploma for admission to higher education
Major: Mathematics and Physics
Thesis: *Rear Suspension Systems for Fullsuspension Bikes* in Physics

WORK EXPERIENCE

since 06/11 **Frontline Media GmbH, Rüti, Switzerland**

Co-Founder, CEO

Frontline Media GmbH runs `http://www.traildevils.ch`, the most visited mountainbike internet platform in Switzerland with more than 4000 daily visitors.

04/11 - 07/11 **CSIR MIAS, Pretoria, South Africa**

Semester Thesis

Development of a navigation and map building algorithm based on RFID tags for an autonomous mining safety platform. Matlab simulation with real world data.

09/09 - 01/10 RUAG Space, Wallisellen, Switzerland

Internship Mechanical design and trade-off study of a measurement and storage unit for biological experiments in zero-gravity condition. CAD design study of novel satellite reflector folding concepts and presentation to European Space Agency.

09/07 – 12/07 Institute of Machine Tools and Manufacturing, ETHZ

Teaching Assistant Teaching and responsibility for 20 students in the exercise lessons of the lecture *Materials and Manufacturing I*.

PEER-REVIEWED JOURNAL PUBLICATIONS

[1] C. Forster, L. Carlone, F. Dellaert, D. Scaramuzza, *"On-Manifold Preintegration for Real-Time Visual-Inertial Odometry,"* IEEE Transactions on Robotics, 2016.
PDF: rpg.ifi.uzh.ch/docs/TRO16_forster_VIO.pdf
Video: youtu.be/CsJkci5lfco

[2] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, D. Scaramuzza, *"Semi-Direct Visual Odometry for Monocular and Multi-Camera Systems,"* IEEE Transactions on Robotics, 2016 (accepted).
PDF: rpg.ifi.uzh.ch/docs/TRO16_forster_SVO.pdf Video: youtu.be/hR8uq1RTUfA

[3] G. Costante, C. Forster, J. Delmerico, P. Valigi, and D. Scaramuzza *"Perception-aware path planning,"* IEEE Transactions on Robotics, 2016 (accepted).
PDF: rpg.ifi.uzh.ch/docs/TRO16_costante.pdf Video: youtu.be/5UmEw8LDJCI

[4] A. Giusti, J. Guzzi, D. Ciresan, F. Lin He, J. P. Rodriguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. A. Di Caro, D. Scaramuzza, L. Gambardella, *"A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots,"* IEEE Robotics and Automation Letters, 2015.
DOI: dx.doi.org/10.1109/LRA.2015.2509024 Video: youtu.be/umRdt3zGgpU

[5] M. Faessler, F. Fontana, C. Forster, E. Mueggler, M. Pizzoli, and D. Scaramuzza, *"Autonomous, vision-based flight and live dense 3D mapping with a quadrotor MAV,"* Journal of Field Robotics, 2015.
DOI: dx.doi.org/10.1002/rob.21581 Video: youtu.be/7-kPiWaFYAc

PEER-REVIEWED CONFERENCE PUBLICATIONS

[6] Z. Zhang, H. Rebecq, C. Forster, D. Scaramuzza, *"Benefit of Large Field-of-View Cameras for Visual Odometry,"* IEEE Int. Conf. on Robotics and Automation, 2016. DOI: dx.doi.org/10.1109/ICRA.2016.7487210 Video: youtu.be/6KXBoprGaRo

[7] C. Forster, L. Carlone, F. Dellaert, D. Scaramuzza, *"IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation,"* Robotics: Science and Systems, 2015. **Best Paper Award Finalist. Oral Presentation: Acceptance Rate 4%.**
DOI: dx.doi.org/10.15607/RSS.2015.XI.006 Video: youtu.be/CsJkci5lfco

[8] C. Forster, M. Faessler, F. Fontana, M. Werlberger, D. Scaramuzza, *"Continuous On-Board Monocular-Vision-based Elevation Mapping Applied to Autonomous Landing of*

Micro Aerial Vehicles," IEEE Int. Conf. on Robotics and Automation, 2015.
DOI: dx.doi.org/10.1109/ICRA.2015.7138988 Video: youtu.be/phaBKFwfcJ4

[9] M. Faessler, F. Fontana, C. Forster, D. Scaramuzza, *"Automatic Re-Initialization and Failure Recovery for Aggressive Flight with a Monocular Vision-Based Quadrotor,"* IEEE Int. Conf. on Robotics and Automation, 2015.
DOI: dx.doi.org/10.1109/ICRA.2015.7139420 Video: youtu.be/pGU1s6Y55JI

[10] E. Mueggler, C. Forster, N. Baumli, G. Gallego, D. Scaramuzza, *"Lifetime Estimation of Events from Dynamic Vision Sensors,"* IEEE Int. Conf. on Robotics and Automation, 2015. DOI: dx.doi.org/10.1109/ICRA.2015.7139876

[11] C. Forster, M. Pizzoli, D. Scaramuzza, *"Appearance-based Active, Monocular, Dense Reconstruction for Micro Aerial Vehicles,"* Robotics: Science and Systems, 2014.
DOI: dx.doi.org/10.15607/RSS.2014.X.029 Video: youtu.be/uAc1pL_c-zY

[12] C. Forster, M. Pizzoli, D. Scaramuzza, *"SVO: Fast Semi-direct Monocular Visual Odometry,"* IEEE Int. Conf. on Robotics and Automation, 2014.
DOI: dx.doi.org/10.1109/ICRA.2014.6906584 Video: youtu.be/2YnIMfw6bJY
Software: `https://github.com/uzh-rpg/rpg_svo`.

[13] M. Pizzoli, C. Forster, D. Scaramuzza, *"REMODE: Probabilistic, Monocular Dense Reconstruction in Real Time,"* IEEE Int. Conf. on Robotics and Automation, 2014.
DOI: dx.doi.org/10.1109/ICRA.2014.6907233 Video: youtu.be/QTKd5UWCGoQ

[14] C. Forster, M. Pizzoli, D. Scaramuzza, *"Air-Ground Localization and Map Augmentation Using Monocular Dense Reconstruction,"* IEEE Int. Conf. on Intelligent Robots and Systems, 2013.
DOI: dx.doi.org/10.1109/IROS.2013.6696924 Video: youtu.be/IZJmZIbinGg

[15] C. Forster, S. Lynen, L. Kneip, D. Scaramuzza, *"Collaborative Monocular SLAM with Multiple Micro Aerial Vehicles,"* IEEE Int. Conf. on Intelligent Robots and Systems, 2013.
DOI: dx.doi.org/10.1109/IROS.2013.6696923 Video: youtu.be/taD3XF2w7Ao

[16] C. Forster, D. Sabatta, R. Siegwart, D. Scaramuzza, *"RFID-Based Hybrid Metric-Topological SLAM for GPS-denied Environments,"* IEEE Int. Conf. on Robotics and Automation, 2013.
DOI: dx.doi.org/10.1109/ICRA.2013.6631324

WORKSHOPS AND LIVE-DEMOS

[17] C. Forster, M. Pizzoli, D. Scaramuzza, *"SVO: Fast Semi-Direct Monocular Visual Odometry,"* Live demonstration at Eur. Conf. on Computer Vision (ECCV), 2014.

[18] C. Forster, S. Lynen, L. Kneip, D. Scaramuzza, *"Collaborative Visual SLAM with Multiple MAVs,"* Robotics: Science and Systems (RSS) Workshop, 2012.
PDF: `http://e-collection.library.ethz.ch/view/eth:7780`